



Titre: Modèle de gestion de la mobilité dans les réseaux IPV6 basé sur les
Title: profils de mobilité des usagers

Auteur: Larbi Alaoui Belrhiti
Author:

Date: 2004

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Alaoui Belrhiti, L. (2004). Modèle de gestion de la mobilité dans les réseaux IPV6
Citation: basé sur les profils de mobilité des usagers [Master's thesis, École Polytechnique
de Montréal]. PolyPublie. <https://publications.polymtl.ca/7330/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7330/>
PolyPublie URL:

**Directeurs de
recherche:**
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

MODÈLE DE GESTION DE LA MOBILITÉ
DANS LES RÉSEAUX IPV6 BASÉ SUR LES
PROFILS DE MOBILITÉ DES USAGERS

LARBI ALAOUI BELRHITI
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
OCTOBRE 2004



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-01272-2

Our file Notre référence

ISBN: 0-494-01272-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MODÈLE DE GESTION DE LA MOBILITÉ
DANS LES RÉSEAUX IPV6 BASÉ SUR LES
PROFILS DE MOBILITÉ DES USAGERS

Présenté par : ALAOUI BELRHITI Larbi

En vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées
a été dûment accepté par le jury d'examen constitué de :

Mme NICOLESCU Gabriela, Doct., Présidente

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. QUINTERO Alejandro, Doct., membre et codirecteur de recherche

M. CHAMBERLAND Steven, Ph.D., membre

REMERCIEMENTS

Tout d'abord, je tiens à remercier mon directeur de recherche, monsieur Samuel Pierre, professeur au département de génie informatique, pour la qualité de son encadrement, sa disponibilité, sa patience, sa compréhension, son encouragement, sa lecture méticuleuse de cette recherche et son support moral et financier. Je tiens à exprimer ma sincère gratitude à mon codirecteur de recherche, monsieur Alejandro Quintero, professeur au département de génie informatique pour sa disponibilité et pour ses suggestions pertinentes. Je tiens à les remercier pour leur dextérité et leur amabilité qui ont fait de ma maîtrise une expérience enrichissante et agréable.

J'adresse aussi mes sincères remerciements aux membres du LARIM (Laboratoire de Recherche en réseautique et Informatique Mobile) de l'École Polytechnique de Montréal, tout spécialement Stéphane Elleingand, Jihad Hodroj et Meral Shirazipour, pour les échanges constructifs que j'ai eus avec eux tout au cours de la réalisation de cette recherche ; et aussi Haikel, Moudji, Othman, Youssef, Gabriel, Djalil et les autres pour leur amitié et leur support moral qui ont fait de mes journées au LARIM une expérience plaisante. Je n'oublierai surtout pas l'extrême gentillesse de Micheline Lafrenière, secrétaire au LARIM ; et Louise Longtin, commis aux dossiers étudiants.

Ce mémoire n'aurait jamais abouti sans le soutien inconditionnel de ma mère que je remercie et que je ne remercierai jamais assez pour tout son dévouement, son affection et son support infailible.

Enfin, je remercie infiniment ma copine pour son amour, son support moral, ses encouragements, et son estime pour le meilleur et pour le pire.

RÉSUMÉ

La gestion de la mobilité dans les réseaux IPv6 permettra aux utilisateurs d'errer à travers les différents réseaux d'accès tout en leur procurant les mêmes services de transfert de données avec une connectivité sans coupure. Sans l'implémentation de Mobile IPv6, des paquets adressés à une unité mobile (MN) ne pourront pas être reçus lorsque l'unité mobile n'est pas dans son lien d'accès d'origine. Mobile IPv6 permet de résoudre ce problème de mobilité en attribuant à chaque unité mobile deux adresses pour la localiser. La première adresse est une adresse fixe appelée *Home Address (HAddr)* représentant l'adresse de l'unité mobile dans son réseau d'origine, l'autre étant une adresse dynamique appelée *Care-Of-Address (COA)* définissant l'emplacement de l'unité mobile dans chaque réseau visité. Les paquets peuvent être routés à cette adresse sans se soucier de l'emplacement de l'unité mobile. À chaque fois que le MN change de COA, il doit donc mettre à jour ses correspondants pour leur donner sa nouvelle adresse dans le nouveau réseau à l'aide d'une association appelée *Binding Update (BU)*. Un des processus, des plus importants et sûrement des plus complexes dans la gestion de la mobilité, est la relève (*Handover* ou *Handoff*), qui permet à l'unité de changer son point d'attache d'un routeur d'accès (*Access Router* ou AR) à un autre. Cette opération nécessite un échange de messages qui peuvent causer l'interruption de la communication. Cette interruption peut se révéler embarrassante pour des applications temps réel ou pour un trafic sensible au délai.

Dans ce mémoire, nous proposons un modèle, *Seamless Mobile IPv6*, pour résoudre le problème de relève sans coupure (*Seamless Handover*), en prenant en compte les profils de mobilité des utilisateurs. Ce modèle a pour but d'offrir une relève sans coupure en tenant compte des profils de mobilité des usagers et de réduire le trafic de signalisation engendré lors de la mise à jour des correspondants. L'étude de performance montre qu'une réduction significative du délai de la relève est obtenue en utilisant le protocole *Seamless Mobile IPv6*. Nous constatons que, indépendamment de

la régularité de l'utilisateur considéré, le protocole *Seamless Mobile IPv6* propose de plus bas délais de relèvement et ne subit aucune perte de paquets durant ses relèvements dans la plupart des cas. Cette étude a aussi montré qu'une réduction importante sur les coûts de la signalisation a été obtenue en utilisant le protocole proposé. En effet, l'usage du patron de mobilité permet d'envoyer un message de mise à jour à ses correspondants dès qu'un changement de réseau est en vue, au lieu d'envoyer une multitude de messages à intervalle défini.

ABSTRACT

Mobility management in IPv6 networks allows a mobile user to move in the network while maintaining all of its present connections and remaining reachable to the rest of the Internet. Without specific support for mobility in IPv6, packets destined to a mobile node would not be able to reach it while the mobile node is away of its home link. In order to continue communication in spite of its movement, a mobile node could change its IP address each time it moves to a new link, but the mobile node would then not be able to maintain transport and higher-layer connection. Mobile IPv6 resolves this problem by allotting the mobile node two addresses. The first address called Home Address, is the mobile node address in its home network. The second address called Care-Of-Address (COA), points to the mobile node in any foreign network and is always changing whenever he changes network. Packets may be routed to the mobile node using its Home Address regardless of the mobile node's current point of attachment to the internet. Whenever the mobile node gets a new COA, he has to update its correspondents by sending them a Binding Update. One of the most important and complex processes in mobility management is the handover process. A handover is the process during which a mobile node switches from one access point to another. This operation cannot happen without sending Binding Updates. During the time when the mobile node is on going a handover, he cannot send or receive packets. This service interruption can be an issue for some real time applications or even some delay sensitive traffics.

In this thesis, we propose a new model called Seamless Mobile IPv6 to solve the handover issue. This new model will deal with this problem using Users Mobility Profiles. Seamless Mobile IPv6 is aimed to reduce the handover latency and the signalization load generated during the location update process. Results analysis showed that the use of Seamless Mobile IPv6 helped us obtain a handover with low delay. In addition, the results showed a significant drop of signalization load using Seamless Mobile IPv6. In fact, better results have been obtained using our protocol in all cases.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES FIGURES.....	xi
LISTE DES TABLEAUX.....	xiii
SIGLES ET ABRÉVIATIONS.....	xiv
LISTE DES ANNEXES.....	xv
 CHAPITRE I - INTRODUCTION	 1
1.1 Définitions et concepts de base.....	1
1.2 Eléments de la problématique	3
1.3 Objectifs de recherche.....	4
1.4 Plan du mémoire	5
 CHAPITRE II - MÉTHODES DE GESTION DE MOBILITÉ DANS LES RÉSEAUX IPV6	 6
2.1 Mobile IPv6.....	6
2.1.1 Terminologie.....	7
2.1.2 Fonctionnement de Mobile IPv6.....	10
2.2 Problèmes avec Mobile IPv6	14
2.3 Hierarchical Mobile IPv6.....	15
2.3.1 Terminologie.....	16
2.3.2 Fonctionnement de Hierarchical Mobile IPv6.....	17
2.4 Fast Handovers for Mobile IPv6.....	19
2.4.1 Terminologie.....	20
2.4.2 Fonctionnement de Fast Handovers for Mobile IPv6	21

2.4.3 Discussion	27
2.5 Cellular IP	28
2.5.1 Fonctionnement de Cellular IP	28
2.5.2 Handoff	29
2.5.3 Synthèse et Discussion.....	30
2.6 Simultaneous Bindings for Mobile IPv6 Fast Handovers.....	31
2.6.1 Terminologie.....	31
2.6.2 Fonctionnement du Simultaneous Bindings	31
2.7 Autres travaux effectués dans le domaine.....	33
2.8 Patron de mobilité	36
2.8.1 Synthèse des méthodes de prédiction du déplacement des usagers	36
 CHAPITRE III - ALGORITHMES ET MODÈLE DE GESTION DE MOBILITÉ.....	39
3.1 Principe de base	39
3.2 Module d'apprentissage des patrons de mobilité	40
3.2.1 Représentation du réseau	41
3.2.2 Patron de mobilité des usagers.....	42
3.2.3 Modèle d'apprentissage du patron de mobilité	43
3.3 Modèle de gestion de mobilité	46
3.3.1 Architecture du modèle de gestion de mobilité	46
3.3.2 Profil de mobilité des usagers	48
3.3.3 Prédiction du moment de la relève.....	50
3.3.4 Formation des adresses IPv6.....	51
3.3.5 Détails de la relève.....	53
3.4 Description des algorithmes.....	55
3.5 Analyse mathématique	61

CHAPITRE IV - IMPLÉMENTATION ET RÉSULTATS	69
4.1 Implémentation du protocole proposé : Seamless Mobile IPv6	69
4.1.1 Correction apportée à MobiWan.....	69
4.1.2 Profil de mobilité	70
4.1.3 Nouveaux paquets.....	70
4.1.4 Modifications apportées au protocole MIPv6.....	71
4.2 Fonctionnement du modèle de gestion de mobilité	74
4.3 Plan d'expérience.....	82
4.4 Analyse des résultats.....	85
 CHAPITRE V - CONCLUSION	 96
5.1 Synthèse des travaux	96
5.2 Limitations des travaux	97
5.3 Indication de travaux futurs	98
 ANNEXE	 99
 BIBLIOGRAPHIE	 113

LISTE DES FIGURES

Figure 2.1 Enregistrement auprès du Home Agent.....	11
Figure 2.2 Routage triangulaire	12
Figure 2.3 Routage optimisé.....	13
Figure 2.4 Domaines MAP dans Hierarchical Mobile IPv6	18
Figure 2.5 Cas d'une configuration Stateless de la Care-Of-Address	23
Figure 2.6 Détails d'une configuration Stateless de la Care-Of-Address	23
Figure 2.7 Détails d'une configuration Stateful de la Care-of-Address	24
Figure 2.8 Messages échangés lors de l'enregistrement.....	26
Figure 2.9 Messages échangés entre l'unité mobile et son HA, le oldAR et le newAR lors du déplacement du MN	27
Figure 3.1 Représentation du réseau cellulaire	41
Figure 3.2 Exemple d'un patron de mobilité	42
Figure 3.3 Modèle d'apprentissage des patrons de mobilité.....	45
Figure 3.4 Architecture répartie	47
Figure 3.5 Architecture centralisée	48
Figure 3.6 Structure d'un identifiant EUI-64.....	52
Figure 3.7 Détails de la relève SMIPv6	54
Figure 3.8 Détails de la relève SMIPv6 échouée	54
Figure 3.9 Détails de la relève MIPv6	55
Figure 3.10 Diagramme d'état de l'unité mobile.....	56
Figure 3.11 Envoi d'un BU pour un nouveau CN et expiration d'un BU	59
Figure 3.12 Réception d'un RA, expiration du profil ou expiration d'une BS.....	61
Figure 3.13 Topologie utilisée pour l'étude analytique	62
Figure 3.14 Détails d'une relève MIPv6.....	63
Figure 3.15 Détails d'une relève FMIPv6.....	64
Figure 3.16 Détails d'une relève SMIPv6.....	66

Figure 4.1 Topologie utilisée pour les scénarios.....	75
Figure 4.2 Profil de mobilité utilisé dans les scénarios.....	75
Figure 4.3 Scénario 1	76
Figure 4.4 Bande passante en fonction du temps – SMIPv6	77
Figure 4.5 Bande passante en fonction du temps – MIPv6.....	77
Figure 4.6 Scénario 2	79
Figure 4.7 Bande passante en fonction du temps – SMIPv6	79
Figure 4.8 Bande passante en fonction du temps – MIPv6.....	80
Figure 4.9 Scénario 3	81
Figure 4.10 Bande passante en fonction du temps – SMIPv6	82
Figure 4.11 Bande passante en fonction du temps – SMIPv6	82
Figure 4.12 Topologie du réseau.....	86
Figure 4.13 Délai de la relève en fonction du délai de la transmission sur Internet	87
Figure 4.14 Nombre de paquets perdus en fonction du délai de la transmission sur Internet	88
Figure 4.15 Signalisation en fonction du délai de la transmission sur Internet	89
Figure 4.16 Signalisation en fonction du délai de la transmission sur Internet	90
Figure 4.17 Nombre de paquets perdus en fonction du débit de paquets	91
Figure 4.18 Nombre de paquets perdus en fonction de la zone de chevauchement	92
Figure 4.19 Bande passante en fonction de la zone de chevauchement	93
Figure 4.20 Délai de la relève en fonction de la vitesse du MN	94
Figure 4.21 Paquets perdus en fonction de la vitesse du MN	95
Figure A.1 Mobilité globale et locale dans NS	102
Figure A.2 Structure d'un nœud de type Mobile IPv4.....	103
Figure A.3 Structure d'un nœud fixe Mobile IPv6 (Class Node)	106
Figure A.4 Structure d'un nœud mobile MIPv6 (Class MobileNode).....	107

LISTE DES TABLEAUX

Tableau 3.1 Exemple d'un historique des déplacements	45
Tableau 3.2 Exemple de profil de mobilité.....	49
Tableau 3.3 Détails chronologiques d'une relève MIPv6.....	63
Tableau 3.4 Détails chronologiques d'une relève FMIPv6.....	65
Tableau 3.5 Détails chronologiques d'une relève SMIPv6.....	66
Tableau 3.6 Comparaison de performance des protocoles MIPv6, FMIPv6 et SMIPv6	67
Tableau 4.1 Niveaux des facteurs	84
Tableau A.1 Simulations du protocole SMIPv6 avec un usager qui suit son profil à 100%	108
Tableau A.2 Simulations du protocole SMIPv6 avec un usager qui suit son profil à 90%	109
Tableau A.3 Simulations du protocole SMIPv6 avec un usager qui suit son profil à 70%	110
Tableau A.4 Simulations du protocole SMIPv6 avec un usager qui suit son profil à 50%	111
Tableau A.5 Simulations du protocole MIPv6.....	111

SIGLES ET ABRÉVIATIONS

AAR	: Anchor Acces Router
ACOA	: Anchor Care-Of-Address
BS	: Base Station
BSC	: Base Station Controllers
CBR	: Constant Bit Rate
DAD	: Duplicate Address Detection
EUI-64	: 64-bit extended unique identifier
FBACK	: Fast Binding Update Acknowledgement
FBU	: Fast Binding Update
FNA	: Fast Neighbor Advertisement
GGSN	: Gateway GPRS Support Node
HACK	: Handover Acknowledge
HI	: Handover Initiate
IEEE	: Institute of Electrical and Electronics Engineers
LCOA	: Local Care-Of-Address
LPS	: Label Switched Path
MAC	: Media Access Control
MPLS	: Multiprotocol Label Switching
MSC	: Mobile Switching Center
NAR	: New Access Router
PAR	: Previous Access Router
PrRtAdv	: Proxy Router Advertisement
RtSolPr	: Router Solicitation for Proxy
UDP	: User Datagram Protocol
VOIP	: Voice Over IP

LISTE DES ANNEXES

Annexe 1 Présentation de Network Simulator NS-2	99
Annexe 1.1 Principes de base	99
Annexe 1.2 Le planificateur.....	101
Annexe 1.3 Les nœuds dans NS.....	101
Annexe 1.4 Modèles de mobilité	103
Annexe 1.5 Autres utilitaires dans NS	104
Annexe 2 MobiWan.....	104
Annexe 2.1 Mobile IPv6 dans MobiWan.....	104
Annexe 2.2 Network Agent	105
Annexe 2.3 Agents Mobile IPv6.....	107
Annexe 3 Résultats des simulations.....	108

CHAPITRE I

INTRODUCTION

Les réseaux cellulaires évoluent très rapidement des réseaux 2G (deuxième génération) actuels vers les réseaux 3G (troisième génération) et plus. Pour suivre cette évolution drastique des réseaux cellulaires et des services Internet d'aujourd'hui, le trafic de données est la prochaine cible du marché. Le protocole IP (*Internet Protocol*) actuel, IPv6, amène ce monde au niveau « Net-Era ». Les deux types de communications, de données et de la voix, dépendent de la technologie IP. Ces tendances dévouent beaucoup d'intérêt pour que les utilisateurs puissent errer à travers les différents réseaux d'accès tout en leur procurant les mêmes services de transfert de données avec une connectivité sans coupure. Les réseaux tout-IP sont des réseaux où IP est utilisé de bout en bout, de l'utilisateur mobile à la passerelle (gateway) qui le connecte à l'Internet. Pour permettre cette évolution, IP doit évidemment permettre la mobilité aux utilisateurs. La gestion de la mobilité IP, objet de ce mémoire, a pour but de procurer aux utilisateurs une connectivité quand ils ne sont pas dans leur réseau d'origine. Dans ce chapitre d'introduction, après avoir défini les concepts de base, nous exposerons les éléments de la problématique et les défis de la recherche qui motivent ce mémoire, présenterons les objectifs visés. Enfin, nous esquisserons le plan de ce mémoire.

1.1 Définitions et concepts de base

Sans l'implémentation de Mobile IPv6, des paquets adressés à une unité mobile (MN) ne pourront pas être reçus lorsque l'unité mobile n'est pas dans son lien d'accès d'origine. Pour pouvoir continuer la communication lors d'un déplacement, une unité mobile devra changer son adresse IP à chaque changement de lien d'accès. Cependant,

l'unité mobile devra perdre les connexions établies au niveau transport et aux couches supérieures quand elle change d'emplacement vu que son adresse IP est différente. En effet, les protocoles qui agissent sur la couche transport (TCP/IP) définissent leur connexion en utilisant les adresses IP des correspondants, d'où la perte de la connexion lors du changement d'IP.

La mobilité sous IPv6 est particulièrement importante, vu qu'une majorité de la population d'Internet sous IPv6 sera constituée d'ordinateurs mobiles [1]. Mobile IPv6 permet de résoudre ce problème de mobilité en attribuant à chaque unité mobile deux adresses pour la localiser. La première adresse étant une adresse fixe appelée *Home Address (HAddr)* représentant l'adresse de l'unité mobile dans son réseau d'origine. L'autre étant une adresse dynamique appelée *Care-Of-Address (COA)* définissant l'emplacement de l'unité mobile dans chaque réseau visité. L'unité mobile va garder la *Home Address* inchangée mais va acquérir une nouvelle *Care-Of-Address* à chaque changement de réseau visité. Ainsi, Mobile IPv6 permet aux unités mobiles de se déplacer d'un lien à un autre sans changer leur *Home Address*. Les paquets peuvent être routés à cette adresse sans se soucier de l'emplacement de l'unité mobile. En effet, une unité se trouvant dans le réseau d'origine de l'unité mobile appelée *Home Agent (HA)* se chargera d'intercepter les paquets destinés à l'unité mobile et de lui expédier à son emplacement actuel en utilisant sa *Care-Of-Address*. A chaque fois que le MN change de COA, il doit donc mettre à jour son *Home Agent* et ses *Correspondent Nodes (CN)* pour leur donner sa nouvelle adresse dans le nouveau réseau à l'aide d'associations entre les deux adresses Haddr - COA appelées *Binding Updates (BU)*. Le déplacement de l'unité mobile de son lien d'origine est donc transparent aux protocoles et applications de la couche transport et des couches supérieures.

Le processus qui permet à l'unité de changer son point d'attache d'un routeur d'accès (*Access Router* ou AR) à un autre est appelé relève (*Handover* ou *Handoff*). Un MN peut subir plusieurs types de relèves : relève au niveau de la couche réseau (*Level 3 Handover* ou *L3 Handover*) et relève au niveau de la couche liaison (*Level 2 Handover*

ou *L2 handover*). Le *L2 Handover* ou relève radio se produit lorsque le MN change de point d'accès radio appartenant au même réseau. Cette relève peut produire un *L3 Handover* ou relève réseau si les points d'accès n'appartiennent pas au même réseau. Dans les cas d'un *L3 Handover*, l'unité mobile doit obtenir une nouvelle adresse IP. Cette opération nécessite un échange de messages qui peuvent causer l'interruption de la communication. La durée de l'interruption peut augmenter avec le nombre d'utilisateurs. Cette interruption peut se révéler embarrassante pour des applications temps réel ou pour un trafic sensible au délai.

Le temps entre le dernier moment où le MN peut recevoir/envoyer des paquets à travers l'ancien routeur d'accès et le premier moment où le MN peut recevoir/envoyer des paquets à travers le nouvel routeur d'accès est appelé Délai de la Relève (*Latency of Handover*). Ainsi, le délai de la relève représente le temps durant lequel le MN ne peut ni recevoir ni envoyer de paquets IP. Ce temps est utilisé pour représenter la performance de la relève.

1.2 Eléments de la problématique

Le protocole Mobile IPv6 fournit sans doute un service de routage et de gestion pour permettre aux unités mobiles d'errer à travers le réseau IP sans perdre la connexion tout en étant localisable par le service. Cependant, ce protocole fait l'œuvre de plusieurs critiques.

En effet, Mobile IPv6 permet aux nœuds de se déplacer dans le réseau Internet tout en restant joignable et en préservant les connexions entre le mobile et les nœuds correspondants. Pour y parvenir, une unité mobile envoie des associations d'adresses (BUs) à son *Home Agent* (HA) et à tous ses *Correspondent Nodes* (CNs) à chaque déplacement d'un sous-réseau à un autre, même si l'unité mobile n'est pas en communication lors de son déplacement. Mobile IPv6 est une solution simple et redimensionnable (scalable) pour la mobilité globale. Cependant, ce n'est pas une bonne solution pour des utilisateurs à haute mobilité. Le coût de la signalisation associé aux

mises à jour d'emplacement peut devenir très important lorsque le nombre de MNs augmente. De plus, si la distance entre le réseau d'origine du MN et le réseau visité et les réseaux des CNs est grande, le délai de signalisation pour la mise à jour d'emplacement sera long.

Par ailleurs, lors de la réception des associations d'adresses (*Binding Updates*) par le *Home Agent* et les *Correspondent Nodes*, il y a une étape d'authentification de ces associations pour vérifier l'expéditeur. L'authentification des BUs demande approximativement une fois et demi le temps que met un paquet pour aller et revenir d'une unité mobile à ses *Correspondent Nodes* pour mettre à jour ces derniers, auquel s'ajoute le temps que met un paquet pour aller et revenir entre l'unité mobile et son HA pour mettre à jour le HA.

La mise à jour du HA peut être faite simultanément avec la mise à jour des CNs. Tous ces délais accumulés représentent un délai total de deux fois et demi le temps que met un paquet entre l'unité mobile et son *Correspondent Node*. Ces délais peuvent interrompre la connexion active à chaque relèvement. Durant ce délai, le MN ne peut ni recevoir ni envoyer de trafic IP à cause du changement de lien et des opérations du protocole MIPv6. Ce délai peut être supérieur au délai minimum acceptable pour un trafic en temps réel ou pour un trafic normal dont la perte de paquet est inacceptable.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est de concevoir un modèle inspiré du protocole Mobile IPv6 pour résoudre le problème de relèvement sans coupure (Seamless Handover), en prenant en compte les profils de mobilité des utilisateurs.

Plus spécifiquement, nous visons les objectifs suivants :

- définir les problèmes de gestion de mobilité auxquels font face les protocoles existants en faisant ressortir leurs faiblesses et leur inadéquation à prendre en compte les patrons de mobilité des usagers;

- proposer un protocole capable d'offrir une relève sans coupure en tenant compte des profils de mobilité des usagers pour réduire le trafic de signalisation nécessité par la recherche et la localisation de ces usagers ;
- implémenter et évaluer la performance de ce nouveau protocole de gestion de mobilité, notamment en le comparant avec les meilleurs protocoles de gestion de mobilité disponibles dans la littérature.

1.4 Plan du mémoire

Ce mémoire est organisé de la manière suivante. Le chapitre 2 analyse les procédures de la gestion de la mobilité dans les réseaux IP et les aspects essentiels qui la caractérisent, ainsi que le protocole Mobile IPv6. Le chapitre 3 propose un nouveau modèle de gestion de la mobilité dans les réseaux IPv6 qui permet de résoudre le problème de relève sans coupure. Au chapitre 4, nous implémentons notre modèle et présentons des résultats qui seront confrontés aux résultats fournis par les meilleurs modèles de gestion de mobilité disponibles dans la littérature. En guise de conclusion, le chapitre 5 résume les principaux résultats obtenus, fait état des limitations des travaux, pour finir avec une indication de recherches futures.

CHAPITRE II

MÉTHODES DE GESTION DE MOBILITÉ DANS LES RÉSEAUX IPV6

Le support de la mobilité dans l'Internet devient de plus en plus crucial surtout que de nombreux équipements mobiles (PDA, téléphones cellulaires) offrant des services d'accès aux réseaux IP font leur apparition (GPRS/WAP). L'apparition de l'UMTS dans le futur proche et son offre de services multimédias impliquent que les réseaux IP devront supporter la mobilité. D'un point de vue topologique, le déplacement d'une unité mobile (MN) dans le réseau se traduit par le changement du point de connexion. Il n'est certainement pas faisable de demander à un utilisateur mobile de changer d'adresse IP à chaque fois qu'il change de localisation. De plus, nous devons garantir la continuité des services et des communications. Le but étant de fournir un service de routage et de gestion pour permettre aux unités mobiles d'errer à travers le réseau IP sans perdre la connexion, tout en étant localisables par le service d'adressage.

Nous détaillerons en premier lieu la mobilité sous IPv6 (MIPv6) et passerons en revue les principaux problèmes de ce protocole. Nous présenterons par la suite des protocoles de mobilité complémentaires à Mobile IPv6. Finalement, nous introduirons le concept de Patron de Mobilité.

2.1 Mobile IPv6

Dans cette partie, nous allons introduire le protocole de gestion de mobilité Mobile IPv6 selon la documentation suivante [1], [2], [3] et [4]. Mobile IPv6 est un protocole développé par l'IETF [1] qui permet aux unités mobiles de rester joignables durant leurs déplacements à travers des réseaux IPv6. Sans l'implémentation de Mobile IPv6, des paquets adressés à une unité mobile (MN) ne pourront pas être reçus lorsque l'unité mobile n'est pas dans son lieu d'accès d'origine. Pour pouvoir continuer la

communication lors d'un déplacement, une unité mobile devra changer son adresse IP à chaque changement de lien d'accès. Cependant, l'unité mobile perdra les connexions établies au niveau transport et aux couches supérieures lors du changement d'emplacement vu que son adresse IP est différente. En effet, les protocoles qui agissent sur la couche transport (TCP/IP) définissent leur connexion en utilisant les adresses IP des correspondants, d'où la perte de la connexion lors du changement d'adresses IP. La mobilité sous IPv6 est particulièrement importante, vu qu'une majorité de la population d'Internet sous IPv6 sera constituée d'ordinateurs mobiles [1].

Mobile IPv6 permet aux unités mobiles de se déplacer d'un lien à un autre sans changer leur adresse d'origine. Les paquets peuvent être routés à cette adresse sans se soucier de l'emplacement de l'unité mobile. L'unité mobile peut aussi continuer à communiquer avec d'autres unités mobiles ou fixes après s'être déplacée à un nouveau lien. Ainsi, le déplacement de l'unité mobile de son lien d'origine est donc transparent aux protocoles et applications de la couche transport et des couches supérieures.

Mobile IPv6 est aussi approprié d'utilisation pour la mobilité dans des réseaux utilisant des technologies d'accès homogènes et hétérogènes. Par exemple, Mobile IPv6 facilite le déplacement d'une unité mobile d'un segment Ethernet à un autre ainsi que d'un segment Ethernet à une cellule WLAN sans changer l'adresse IP.

On pourrait donc voir le protocole Mobile IPv6 comme résolvant le problème de la gestion de mobilité au niveau de la couche réseau. Les problèmes de la gestion de mobilité au niveau de la couche d'accès étant déjà résolus. Par exemple, dans plusieurs solutions WLAN, nous retrouvons des mécanismes de mobilité au niveau accès qui permettent la relève d'une cellule à une autre en rétablissant la connectivité au niveau accès.

2.1.1 Terminologie

- **MN (*Mobile Node*)** : Un nœud/unité mobile qui peut changer son point d'attache d'un lien à un autre tout en étant joignable à travers son adresse d'origine.

- **CN (*Correspondent Node*)** : Un autre nœud avec qui le MN communique. Le CN peut-être soit un nœud/unité fixe ou mobile.
- **HA (*Home Agent*)** : Un routeur sur le lien d'origine d'une unité mobile où est enregistré l'adresse d'origine (*Home Address*) de cette unité. Quand l'unité mobile ne sera pas sur son lien d'origine, le *Home Agent* va intercepter les paquets sur le lien d'origine destiné à l'adresse d'origine de l'unité mobile, les encapsuler et les envoyer à travers un tunnel à la *Care-Of-Address* (définie plus bas) de l'unité mobile.
- **Home Address (*HAddr*)** : Une adresse permanente associée à une unité mobile comme étant son adresse d'origine. Cette adresse représente l'emplacement de l'unité mobile dans son réseau d'origine. Cette adresse fait partie du lien d'origine de l'unité mobile. Lors du routage IP, les paquets destinés à l'adresse d'origine de l'unité mobile seront acheminés sur son lien d'origine.
- **COA (*Care-Of-Address*)** : Une adresse unicast temporaire associée à une unité mobile lors d'un déplacement sur un lien visité. Une unité mobile peut avoir plusieurs COA enregistrées simultanément avec plusieurs routeurs d'accès. Parmi toutes ces COA, celle enregistrée auprès du HA est appelée *Primary Care-Of-Address*.
- **Lien** : un moyen de communication à travers lequel les nœuds peuvent communiquer au niveau de la couche liaison (couche 2), comme Ethernet.
- **Lien d'origine** : le lien sur lequel le préfixe sous-réseau du réseau d'origine du MN est défini.
- **Lien visité** : c'est tout lien qui ne se trouve pas dans le réseau d'origine du nœud mobile.
- **Relève à la couche 2 (*L2 handover*)** : Un processus par lequel un MN va changer sa connexion de la couche d'accès pour une autre. Par exemple, un changement de point d'accès dans un réseau sans fil est un *L2 Handover*.

- **Relève à la couche 3 (L3 handover)** : Succède au *L2 handover*, lorsqu'une unité mobile détecte une différence du préfixe du sous-réseau et qu'un changement de la COA devra se faire. Par exemple, un changement de routeur d'accès qui n'appartienne pas au même réseau est un *L3 handover*.
- **Association (Binding)** : Une association de l'adresse *HAddr* et de l'adresse COA pour une unité mobile ainsi que la durée de vie de cette association.
- **Enregistrement (Registration)** : Le processus par lequel une unité mobile envoie une mise à jour des associations à son HA ou CN entraînant l'enregistrement d'une association (*Binding*) pour cette unité mobile.

Mobile IPv6 a besoin de nouveaux messages pour certaines procédures n'existant pas sous IPv6. Ces messages sont définis dans le protocole IPv6 comme des *IPv6 Destination Options* :

- **Mise à jour d'une association (Binding Update ou BU)** : Une mise à jour des associations est utilisée par un MN pour aviser un CN ou HA de sa nouvelle COA et envoyant une nouvelle association.
- **Confirmation d'une association (Binding Acknowledgement ou BU Ack)** : La confirmation d'une association est utilisée comme accusé de réception d'un BU. Ce dernier n'est envoyé que si un accusé de réception est demandé lors de l'envoi du BU.
- **Demande de Mise à jour de l'association (Binding Refresh Request ou BR)** : Une demande de mise à jour de l'association est utilisée par un CN pour demander au MN de mettre à jour son association. Ce message est souvent utilisé quand l'association d'un MN arrive à expiration.

Mobile IPv6 utilise les structures de données conceptuelles suivantes :

- **Binding Cache** : Pour pouvoir communiquer avec les autres nœuds, il est nécessaire de garder une liste de correspondance appelée *Binding Cache*. Cette liste est mise à jour à chaque fois qu'un nouveau BU est reçu. Avant de

transmettre un paquet, une unité mobile vérifie l'existence d'une entrée dans le *Binding cache*.

- ***Binding Update List*** : Cette liste est maintenue par chaque unité mobile. Elle contient des informations concernant chaque association que ce MN a fait ou est en train de faire avec un autre nœud. Ces informations sont supprimées quand la durée de vie du BU expire.
- ***Home Agents List*** : Les *Home Agents* ont besoin de savoir quels autres *Home Agents* se trouvent sur le même lien. Cette information est conservée dans une liste de type *Home Agents List*. Cette liste est utilisée pour informer les MN durant la découverte dynamique des adresses des HA.

2.1.2 Fonctionnement de Mobile IPv6

Une unité mobile doit toujours être joignable à travers sa *Home Address* (*HAddr*), même si ce MN est attaché à son lien d'origine ou est en déplacement. Quand une unité mobile est dans son lien d'origine, les paquets adressés à son *HAddr* vont être routés au lien d'origine de l'unité mobile en utilisant un mécanisme de routage Internet traditionnel.

Lorsque l'unité mobile est sur un lien visité, elle est donc joignable à travers sa *Care-Of-Address* (COA). Une unité mobile peut obtenir une COA grâce à des mécanismes conventionnels de IPv6, comme *Stateless* ou *Stateful Auto-Configuration*. Pendant que l'unité mobile reste à cet emplacement, les paquets adressés à cette COA seront routés à l'unité mobile.

L'association entre la *Home Address* et la *Care-Of-Address* d'une unité mobile est appelée « *Binding* ». Pendant un déplacement en dehors du lien d'origine, une unité mobile enregistre sa *Care-Of-Address* primaire avec un routeur sur son lien d'origine, en demandant à ce routeur d'être son *Home Agent*. L'unité mobile effectue ce *Binding Registration* en envoyant un message de type *Binding Update* au *Home Agent*. Le *Home Agent* répond à l'unité mobile en envoyant un message de type *Binding*

Acknowledgement. L'enregistrement auprès du Home Agent est illustré par la Figure 2.1.

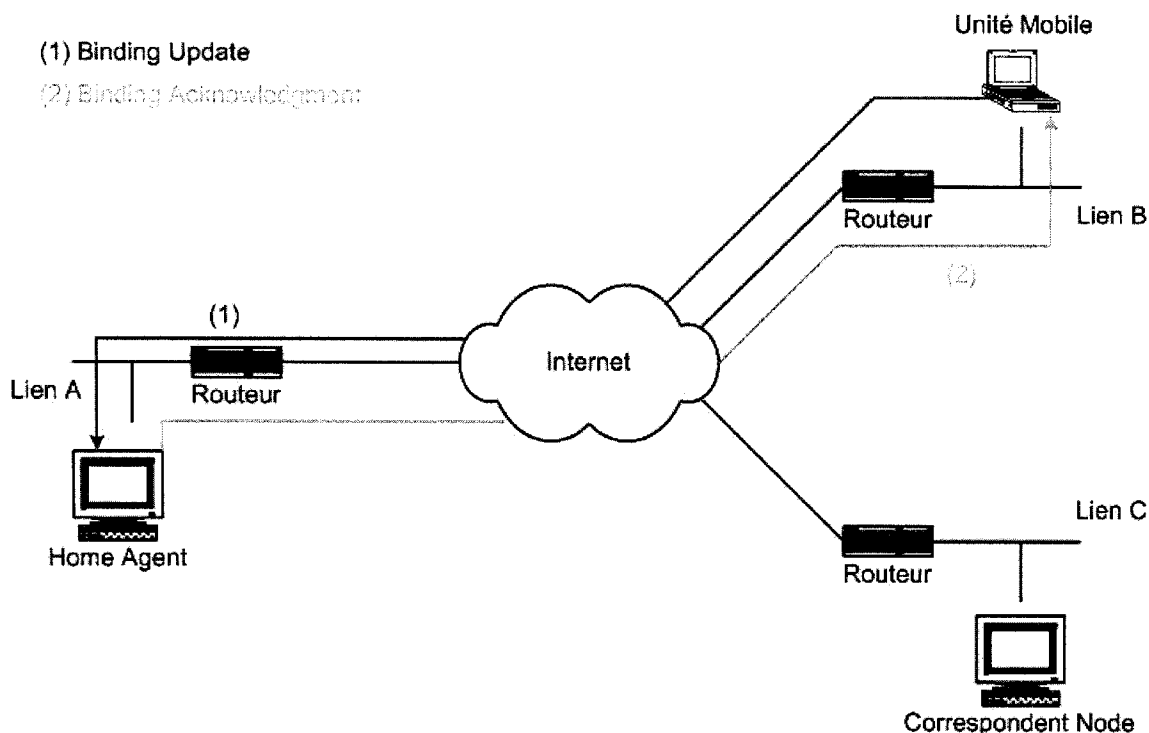


Figure 2.1 Enregistrement auprès du Home Agent

N'importe quel nœud communiquant avec une unité mobile est appelé *Correspondent Node* et peut être mobile ou fixe. Les unités mobiles peuvent fournir des informations sur leur emplacement actuel aux *Correspondent Nodes*. Ce processus est appelé *Correspondent Registration*. Durant ce processus, un test appelé *Return Routability Test* est effectué pour autoriser l'établissement de l'association.

On distingue deux méthodes pour qu'une unité mobile et son *Correspondent Node* puissent communiquer : le routage triangulaire et le routage optimisé.

La première méthode, appelée méthode de routage triangulaire, étant l'utilisation d'un tunnel bidirectionnel, n'a pas besoin que le CN supporte Mobile IPv6 et peut être utilisée même si le MN n'a pas enregistré son association auprès du CN. Les paquets provenant du CN sont routés au *Home Agent*. Par la suite, ces paquets seront envoyés à

travers le tunnel entre le HA et le MN. Les paquets envoyés par le MN pour les CN sont routés normalement en utilisant l'Internet. Dans cette méthode, le *Home Agent* utilise *Proxy Neighbor Discovery* pour intercepter n'importe quel paquet IPv6 adressé à l'unité mobile (*Home Address*) sur le lien d'origine. Le routage triangulaire est illustré par la Figure 2.2.

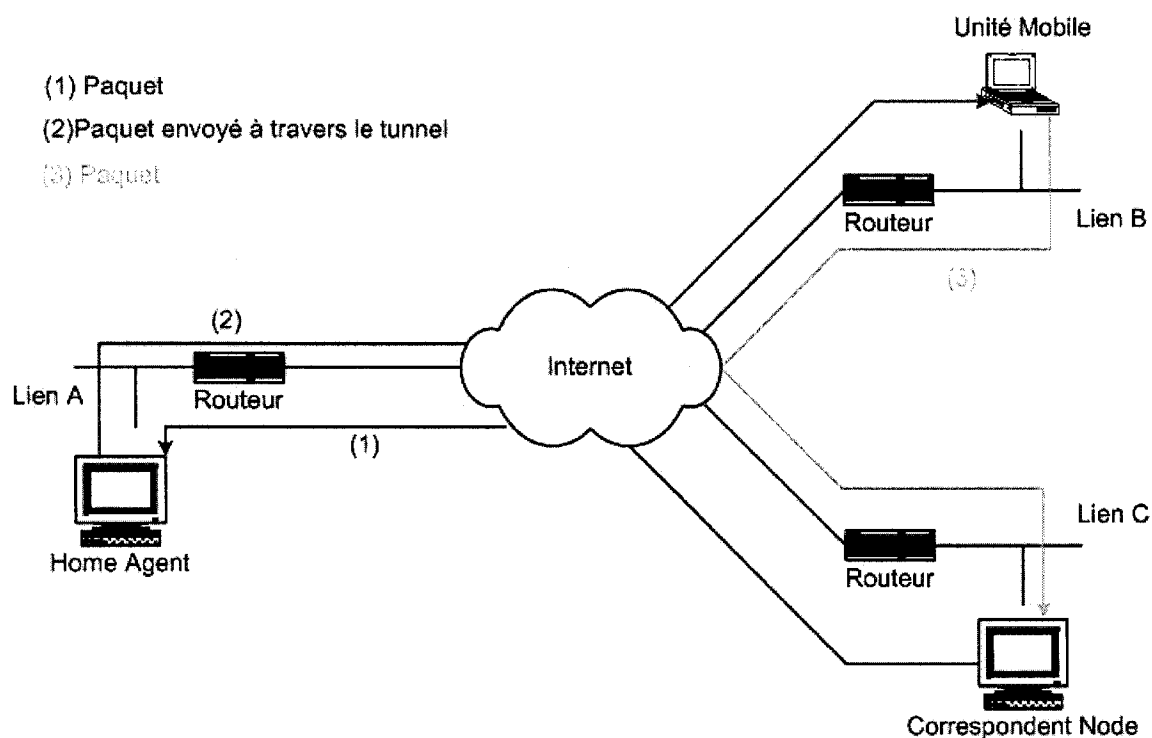


Figure 2.2 Routage triangulaire

La deuxième méthode, Routage Optimisé, requiert que l'unité mobile enregistre son association auprès du *Correspondent Node* (voir Figure 2.3). Ainsi, les paquets envoyés par le *Correspondent Node* peuvent être routés directement à la COA du MN. Quand un paquet est envoyé par le CN à n'importe quel MN, le *Correspondent Node* vérifie d'abord son *Binding Cache* pour trouver l'association d'adresses du MN en question. Si l'information se trouve dans le *Binding Cache*, le nœud utilise un nouveau type d'entête de routage IPv6 pour router le paquet au MN en utilisant la *Care-Of-Address* indiquée dans l'association.

Le fait de router les paquets directement à la COA de l'unité mobile permet l'utilisation d'un plus court chemin. Cela permet aussi d'éliminer la congestion au niveau du *Home Agent* du MN et du lien d'origine. De plus, l'impact d'une panne du *Home Agent* ou des réseaux sur le chemin du MN est réduit.

En routant les paquets directement au MN, le CN met la *Care-Of-Address* du MN comme l'adresse du destinataire dans l'entête IPv6. Un nouveau genre d'entête a été ajouté au paquet pour spécifier la *Home Address*. De la même manière, le MN spécifie l'adresse source dans l'entête IPv6 du paquet comme étant sa *Care-Of-Address*. Le MN ajoute une nouvelle option IPv6 de type *Home Address Destination Option* pour contenir sa *Home Address*. L'inclusion du *Home Address* dans les paquets rend l'utilisation de la *Care-Of-Address* transparente au niveau de la couche réseau.

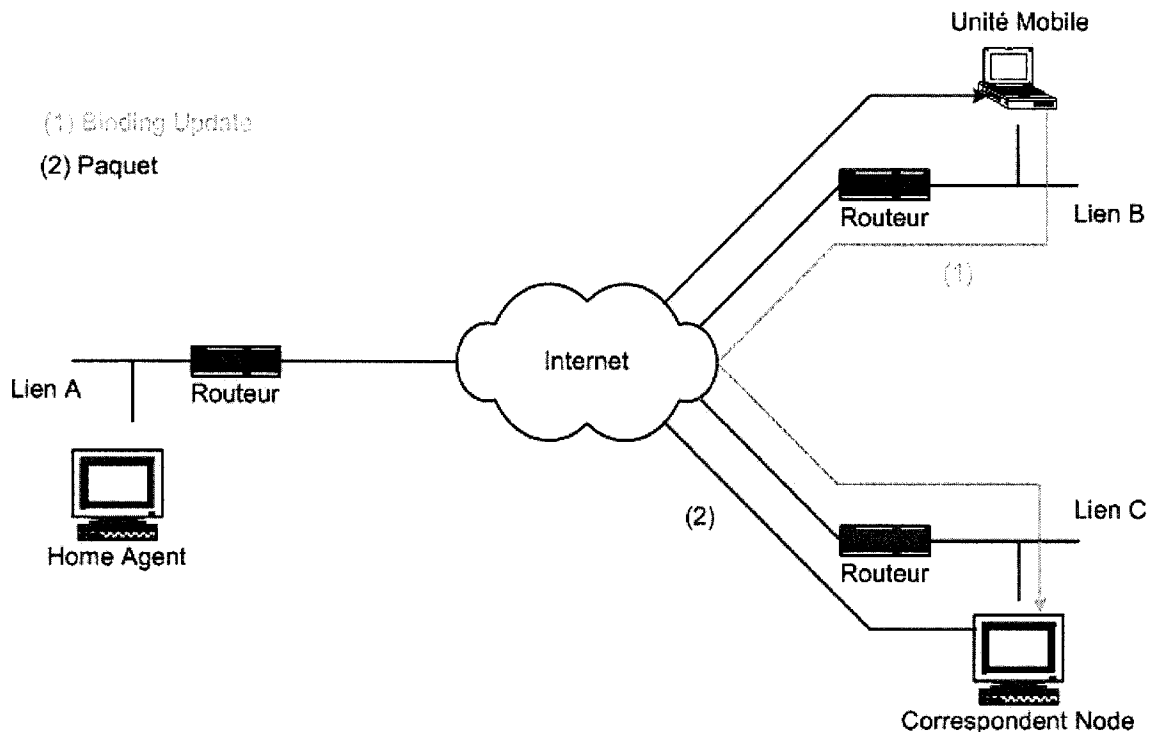


Figure 2.3 Routage optimisé

Mobile IPv6 aussi offre un support pour plusieurs *Home Agents* et un support limité pour la reconfiguration du réseau d'origine. Dans ces cas, le MN peut ne pas connaître l'adresse de son *Home Agent*, et même le préfixe de son sous-réseau d'origine peut changer au cours du temps. Un mécanisme, connu sous le nom de *Dynamic Home Agent Address Discovery* permet au MN de découvrir dynamiquement l'adresse IP d'un *Home Agent* sur son lien d'origine, même quand le MN n'est pas sur son lien d'origine. Les unités mobiles peuvent avoir des informations sur leur préfixe du sous-réseau d'origine grâce à un mécanisme appelé *Mobile Prefix Discovery*.

2.2 Problèmes avec Mobile IPv6

Le protocole Mobile IPv6 fournit certes un service de routage et de gestion pour permettre aux unités mobiles d'errer à travers le réseau IP sans perdre la connexion tout en étant localisable par le service. Cependant, ce protocole fait l'œuvre de plusieurs critiques. En effet, Mobile IPv6 permet aux nœuds de se déplacer dans le réseau Internet tout en restant joignable et en préservant les connexions entre le mobile et les nœuds correspondants. Pour y parvenir, une unité mobile envoie des associations d'adresses (BU) à son *Home Agent* (HA) et à tous ses *Correspondent Nodes* (CN) à chaque déplacement d'un sous-réseau à un autre, même si l'unité mobile n'est pas en communication lors de son déplacement. Mobile IPv6 est une solution simple et évolutive (scalable) pour la mobilité globale. Cependant, ce n'est pas une bonne solution pour des utilisateurs à haute mobilité. Le coût de la signalisation associé aux mises à jour d'emplacement peut devenir très important lorsque le nombre de MN augmente. De plus, si la distance entre le réseau visité et le réseau d'origine du MN ou la distance entre le réseau visité et les réseaux des CN est grande, le délai de signalisation pour la mise à jour d'emplacement sera long.

Par ailleurs, lors de la réception des associations d'adresses (*Binding Updates*) par le *Home Agent* et les *Correspondent Nodes*, il y a une étape d'authentification de ces associations pour vérifier l'identité de l'expéditeur. L'authentification des BU demande

approximativement une fois et demi le temps que met un paquet pour aller et revenir d'une unité mobile à ses *Correspondent Nodes* pour mettre à jour ces derniers, auquel s'ajoute le temps que met un paquet pour aller et revenir entre l'unité mobile et son HA pour mettre à jour le HA.

La mise à jour du HA peut être faite simultanément avec la mise à jour des CN. Tous ces délais accumulés représentent un délai total de deux fois et demi le temps que met un paquet entre l'unité mobile et son *Correspondent Node*. Ces délais peuvent interrompre la connexion active à chaque relève. Durant ce délai, le MN ne peut ni recevoir ni envoyer de trafic IP à cause du changement de lien et des opérations du protocole MIPv6. Ce délai peut être supérieur au délai minimum acceptable pour un trafic en temps réel ou pour un trafic normal dont la perte de paquet est inacceptable.

2.3 Hierarchical Mobile IPv6

Hierarchical Mobile IPv6 (HMIPv6) est un protocole complémentaire à Mobile IPv6 développé par l'IETF [5] pour faciliter la gestion de la mobilité locale. En effet, HMIPv6 introduit une architecture plus hiérarchisée pour diminuer la signalisation globale et permettre une meilleure gestion de la mobilité locale. Une gestion de la mobilité hiérarchique pour MIPv6 réduit le poids de la signalisation entre l'unité mobile, ses *Correspondent Nodes* et son *Home Agent*. En effet, le fait que le réseau soit découpé en différents domaines gérés par un *Mobility Anchor Point* (MAP) fait qu'une unité mobile n'ait pas besoin de mettre à jour ses correspondants lors d'un déplacement dans le même domaine. D'un autre côté, le *Mobility Anchor Point* (MAP) peut être utilisé pour améliorer la performance de MIPv6 en termes de délai de relève (*Handoff Latency*).

Le *Mobility Anchor Point* est un nouveau nœud de type IPv6 qui se situe à n'importe quel niveau dans un réseau hiérarchique de routeurs, incluant les *Access Routers* (AR). Contrairement au *Foreign Agents* dans Mobile IPv4, un MAP n'est pas nécessaire dans chaque sous-réseau. Le MAP va diminuer la quantité de signalisation de

Mobile IPv6 en dehors du domaine local. L'introduction du MAP résout plusieurs problèmes cités auparavant de la manière suivante :

- l'unité mobile envoie des mises à jour des associations au MAP au lieu d'en envoyer au HA et aux CN qui peuvent être assez loin ;
- un seul BU doit être envoyé par le MN pour que le trafic provenant du HA et des CN soit re-routé à son nouvel emplacement.

Un MAP est essentiellement un *Home Agent* local. Le but d'un modèle hiérarchique pour la gestion de la mobilité dans Mobile IPv6 est d'améliorer la performance de celui-ci tout en minimisant l'impact sur Mobile IPv6 et d'autres protocoles IPv6. Il permet aussi l'utilisation de *Fast Handovers for Mobile IPv6* pour aider les unités mobiles à jouir d'une mobilité sans coupure. De plus, HMIPv6 permet aux unités mobiles de laisser leur emplacement inconnu des autres *Correspondent Nodes* et *Home Agents* tout en utilisant le routage optimisé de Mobile IPv6.

2.3.1 Terminologie

- **Routeur d'accès (Access Router ou AR)** : Le routeur par défaut à travers lequel l'unité mobile communique. Le AR traite tout le trafic sortant des MN.
- **Mobility Anchor Point (MAP)** : Un *Mobility Anchor Point* est un routeur qui se trouve dans un réseau visité par l'unité mobile. Le MAP est utilisé par le MN comme un HA local. Un ou plusieurs MAP peuvent exister dans un réseau visité
- **Regional Care-of-Address (RCOA)** : Une RCOA est une adresse obtenue par l'unité mobile dans le réseau visité. Elle est auto-configurée par l'unité mobile quand elle reçoit les options du MAP. Dans MIPv6, cette adresse représente la COA simplement et peut donc être utilisée pour rejoindre l'unité mobile lors d'un déplacement.
- **On-link CoA (LCOA)** : La LCOA est configurée sur l'interface de l'unité mobile en utilisant le préfixe annoncé par le routeur d'accès du réseau visité. Cette adresse sera utilisée seulement par le routeur pour expédier les paquets à

l'unité mobile. Cependant, la LCOA est utilisée pour la différencier du RCOA dans le sens où la LCOA est simplement une adresse locale et n'est pas connue du HA et des CN.

- **Local Binding Update :** Le MN envoie un *Local Binding Upadate* au MAP pour établir une association (*Binding*) entre la RCOA et la LCOA.

2.3.2 Fonctionnement de Hierarchical Mobile IPv6

Hierarchical Mobile IPv6 introduit une fonction, le *Mobility Anchor Point* (MAP), et quelques extensions mineures aux opérations des unités mobiles. Le fonctionnement du *Correspondent Node* et du *Home Agent* ne change pas.

Un MN qui pénètre un domaine MAP recevra des *Router Advertisements* contenant de l'information sur un ou plusieurs MAP locaux. Le MN peut faire un *binding* associant son emplacement actuel (*on-link COA*) avec une adresse appartenant au sous-réseau du MAP (RCOA). Se comportant comme un *Home Agent* local, le MAP recevra tous les paquets adressés au MN utilisant la RCOA, les encapsulera et les enverra directement à l'adresse courante (LCOA) du MN. Si le MN se déplace dans le domaine MAP, il lui suffira d'enregistrer sa nouvelle adresse (LCOA) auprès du MAP.

Cependant, seule le changement de l'adresse régionale COA (RCOA) a besoin d'être enregistré auprès du HA et des CN. La RCOA ne change pas lorsque le MN se déplace dans un domaine MAP, ce qui rend la mobilité transparente aux CN avec lequel le MN communique. Un domaine MAP est délimité par les routeurs d'accès qui publient les informations du MAP. La Figure 2.4 illustre l'utilisation du MAP dans un réseau visité.

A la Figure 2.4, le MAP peut procurer une mobilité sans coupure quand le déplacement se fait de deux *Access Routers* attachés au même MAP. Une hiérarchie

multi-niveaux n'est pas nécessaire pour une meilleure performance de relève. Cependant, il est suffisant de positionner un ou plusieurs MAP (possiblement dans le même domaine) à n'importe quel endroit dans le réseau.

A son arrivée dans un réseau visité, l'unité mobile va découvrir l'adresse globale du MAP. Cette adresse est conservée dans les *Access Routers* et sera communiquée aux nœuds mobiles à travers des *Router Advertisements* (RA). Ces *Advertisements* sont nécessaires pour informer les MN de la présence du MAP (MAP *Discovery*). La phase de découverte (*Discovery*) informera aussi le MN de la distance entre le MAP et le MN.

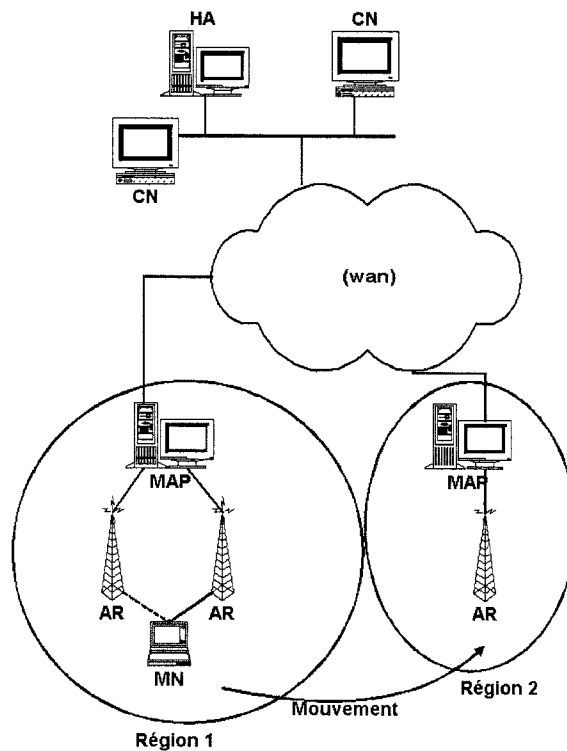


Figure 2.4 Domaines MAP dans Hierarchical Mobile IPv6

Le processus de MAP *Discovery* se poursuit pendant que le MN se déplace d'un sous-réseau à un autre. Si le MN erre dans un domaine MAP, les AR sont configurés

pour annoncer la même adresse MAP. S'il y a une différence d'adresse MAP publiée, le MN doit effectuer une détection de mouvement et envoyer les BU à son HA et CN.

Le MN va d'abord s'enregistrer avec un MAP en lui envoyant un BU contenant sa *Home Address* et la *On-Link address* (LCOA). La *Home Address* utilisée dans le BU est la RCOA. Le MAP doit conserver les informations dans son *Binding Cache* pour être capable de rediriger les paquets à leur destination finale. L'unité mobile aura toujours besoin de savoir l'expéditeur des paquets pour effectuer le routage optimisé. Cette information sera disponible pour le MN puisque le MAP ne modifie pas le contenu du paquet. Pour mieux utiliser la bande passante du réseau, un MN peut décider de s'enregistrer avec plusieurs MAP en même temps et utiliser chaque adresse MAP pour un groupe spécifique de CN.

En effet, il y a quelques points sur lesquels HMIPv6 est critiqué. Ainsi, mis à part les améliorations qu'apporte HMIPv6 à MIPv6, son évolutivité n'est pas optimum. Le point faible de HMIPv6 est le routage du trafic entrant au niveau du MAP. L'optimisation du routage est sacrifiée pour obtenir une plus faible signalisation. Dans un réseau public, le MAP peut devenir un goulot d'étranglement et détériorer la qualité de service. Avec un grand volume de trafic, ce routage indirect du trafic avec HMIPv6 peut créer des congestions. D'un autre côté, HMIPv6 permet bien de réduire la signalisation de la gestion de mobilité au détriment d'autres problèmes mais ne résout pas vraiment le problème de relève. En effet, les relèves ne génèrent plus autant de trafic mais ne sont pas sans coupure. Il est vrai qu'une relève dans une région MAP est presque sans coupure, mais une relève inter-régions est inchangée.

2.4 Fast Handovers for Mobile IPv6

Dans cette partie, nous allons introduire le protocole *Fast Handovers for Mobile IPv6* à partir de [6] et [7]. Le but du protocole *Fast Handovers for Mobile IPv6* (FMIPv6), développé par l'IETF [6], est de permettre au MN de configurer une nouvelle *Care-Of-Address* avant de changer de sous-réseau, pour être capable de l'utiliser juste

après la connexion au nouvel Access Router. La solution de FMIPv6 est de minimiser le délai pendant la relève où le MN ne peut ni envoyer ni recevoir de paquets (*Handover Latency*). Le principe est d'établir une nouvelle COA avant de perdre la connexion avec l'ancien *Access Router* (*oldAR*). Ensuite, quand le MN est attaché à un nouvel *Access Router* (*newAR*), il peut continuer ses communications avec sa nouvelle adresse qui est déjà connue. Si l'enregistrement prématuré échoue, le MN peut toujours effectuer une relève traditionnelle. De plus, le *Fast Handover* met en place un système d'expédition des paquets entre le *oldAR* et le *newAR*. L'établissement d'une nouvelle COA avant que le MN ne se déplace nécessite de l'anticipation sur les déplacements du MN. Cette anticipation peut être faite grâce aux messages échangés au niveau physique ou simplement à l'aide d'informations pertinentes au niveau 2 (mesure du signal, etc.). Le but est d'aboutir à une relève au niveau 3 (*L3 Handover*) avant que la relève au niveau 2 (*L2 Handover*) n'ait pris fin.

2.4.1 Terminologie

- ***Old Access Router (oldAR)*** : Le AR qui est chargé de manipuler le trafic du MN avant le *L2 handover*. Le *oldAR* est le routeur auquel le MN est actuellement attaché (ou routeur par défaut).
- ***New Access Router (newAR)*** : Le AR qui est chargé de manipuler le trafic du MN après le *L2 handover*. Le *newAR* est le routeur auquel le MN compte s'enregistrer.
- ***Anchor Access Router (AAR)*** : Le AR où le MN a enregistré sa COA originellement (i.e., la COA qui est maintenant connue par le HA et les CN).
- ***Anchor Care-Of-Address (aCOA)*** : La COA à laquelle le HA et les CN expédient les informations destinées au MN.
- ***Old Care-Of-Address (oldCOA)*** : La COA qui était valide sur le *oldAR*. Le MN peut réutiliser cette adresse pendant son enregistrement au *newAR* en attendant de finir ses opérations.

- ***New Care-Of-Address (newCOA)*** : La COA qui est valide sur le newAR.

FMIPv6 a besoin de nouveaux messages pour certaines procédures n'existant pas sous IPv6. Ces messages sont définis pour la communication entre les AR et aussi entre les AR et le MN.

- ***Router Solicitation for Proxy (RtSolPr)*** : Un message envoyé par le MN au *oldAR* pour demander des informations à propos d'une relève potentielle.
- ***Proxy Router Advertisement (PrRtAdv)*** : Un message du *oldAR* au MN pour lui dire qu'il peut entamer une relève.
- ***Fast Binding Update (FBU)*** : Un message du MN au *oldAR* pour lui demander d'expédier son trafic au *newAR*.
- ***Fast Neighbor Advertisement (FNA)*** : un message du MN au *newAR* pour confirmer l'utilisation du *newCOA* quand le MN n'a pas reçu de FBACK.
- ***Handover Initiate (HI)*** : un message du *newAR* au *oldAR* pour initier une relève.
- ***Handover Acknowledge (HACK)*** : Un message du *newAR* au *oldAR* confirmant le HI.

2.4.2 Fonctionnement de Fast Handovers for Mobile IPv6

Dans cette section, nous allons considérer que le *newAR* est connu par le *oldAR*, le *newAR* accepte que le MN s'attache au nouveau sous-réseau et que la COA proposée soit valide. Dans le cas contraire, le *Fast Handover* échoue et le MN peut effectuer une relève à la manière de MIPv6. La structure décrite précédemment s'applique particulièrement aux deux scénarios suivants :

- La relève initiée par le réseau :

Le réseau décide de la relève (le *oldAR* décide du nouveau point d'attache pour le MN). Le *oldAR* commence par envoyer des signaux au MN et au *newAR* pour commencer un L3 Handover.

- La relève initiée par le Mobile :

Le MN décide de la relève (le mobile doit détecter et initier la relève). Le MN peut imposer son choix de déplacement à un nouveau point d'attache. Le MN commence par envoyer des signaux au *oldAR* pour commencer la relève.

2.4.2.1 La relève initiée par le réseau

Lors d'une relève initiée par le réseau, le *oldAR* reçoit une indication montrant que le MN va se déplacer ainsi que les informations sur le *newAR* auquel le MN va s'attacher. La *newCOA* peut être configurée de manière *stateless* ou *stateful*.

- **Cas d'une configuration Stateless de la Care-of-Address**

Quand le *oldAR* s'aperçoit que le MN va se déplacer à un nouveau AR, le *oldAR* prépare une *newCOA* en utilisant l'adresse MAC du mobile et le préfixe du *newAR*. Ensuite, cette *newCOA* est envoyée au MN avec l'adresse IP du *newAR* et l'adresse de la couche liaison en utilisant des messages *Proxy Router Advertisement* (PrRtAdv). Au même moment, le *oldAR* envoie un HI au *newAR* pour lui indiquer l'ancienne et la nouvelle COA du MN. Le *newAR* vérifie si la *newCOA* est valide en s'assurant qu'elle n'est pas utilisée par quelqu'un d'autre. Si la COA est valide et acceptable par le *newAR*, alors le *newAR* ajoute l'adresse dans sa cache visiteur pour un moment et répond avec un HAck. Si l'adresse n'est pas valide, le *newAR* répond toujours au *oldAR* avec un Hack en spécifiant que le Handover est accepté mais que l'adresse n'est pas valide. Les Figures 2.5 et 2.6 illustrent l'échange de messages lors de l'initiation de la relève.

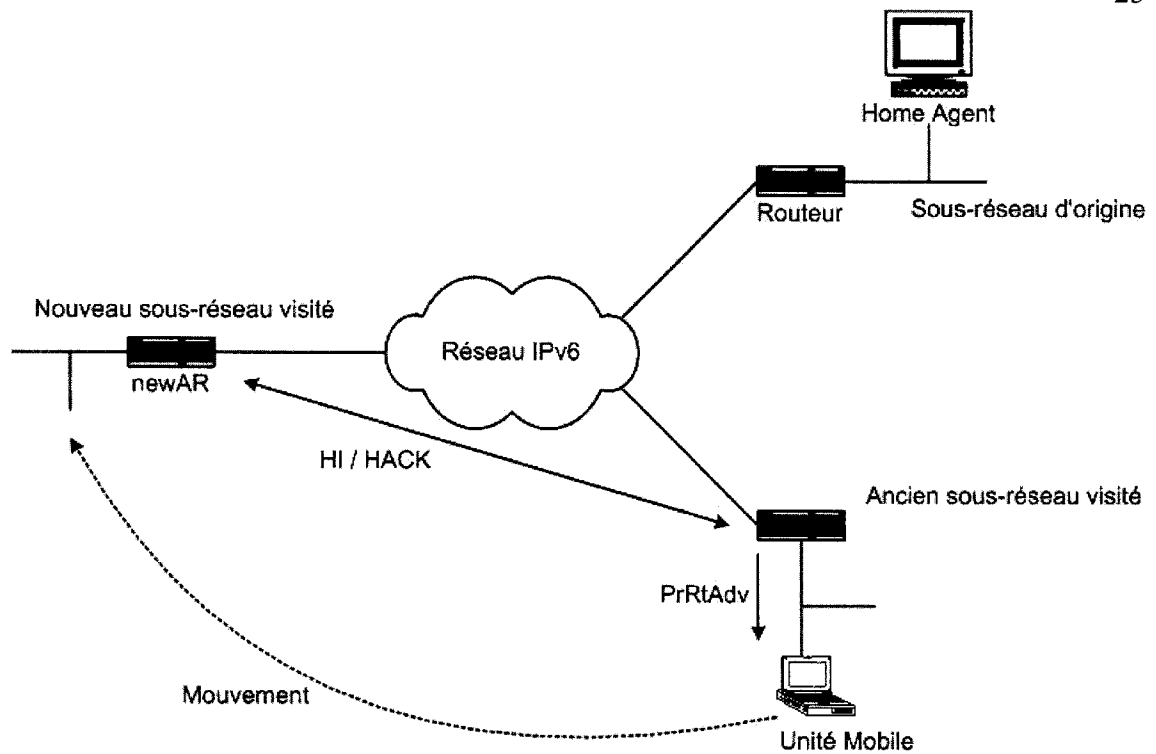


Figure 2.5 Cas d'une configuration Stateless de la Care-Of-Address

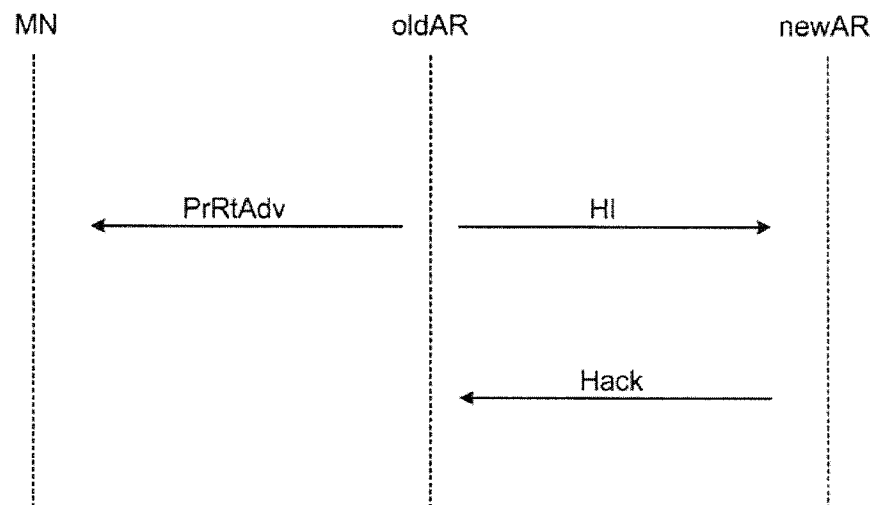


Figure 2.6 Détails d'une configuration Stateless de la Care-Of-Address

- **Cas d'une configuration Stateful de la Care-of-Address**

Dans le cas d'une configuration d'adresse *stateful*, le *oldAR* doit envoyer un HI avant le *Proxy Router Advertisement* (PrRtAdv). Le HI est utilisé pour demander une nouvelle adresse temporaire pour le mobile et non pas pour en valider une. Le *Handover Acknowledgement* (HAck) contient une adresse valide qui peut être transmise au mobile dans le *Proxy Router Advertisement*. Le nouveau AR échange des messages avec le *oldAR* et expédie les messages entre le *oldAR* et le MN. Quand le *newAR* reçoit le HI envoyé par le *oldAR* sans COA, il alloue une nouvelle COA et l'envoie au *oldAR* à travers un HAck. Si le *newAR* reçoit un HI avec une nouvelle COA, il détermine si cette *newCOA* est valide et envoie une validation dans le HAck. L'échange de ces HI/HAck précède l'envoi du PrRtAdv du *oldAR* au MN et le reste du processus est identique à la configuration *stateless*.

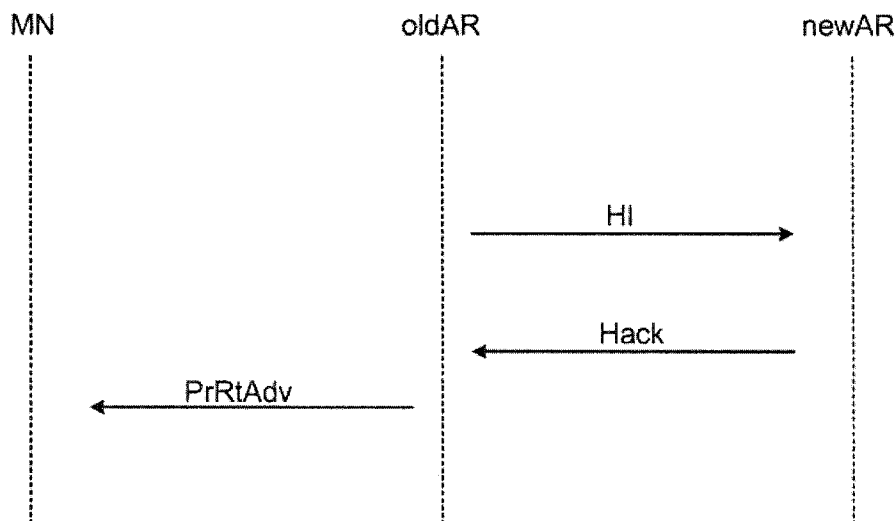


Figure 2.7 Détails d'une configuration Stateful de la Care-of-Address

2.4.2.2 La relève initiée par le mobile

La plus grande différence entre la relève initiée par le mobile et celle initiée par le réseau est que, dans la relève initiée par le mobile, le MN reçoit des informations sur le L2 Handover. Le MN doit envoyer un *RtSolPr* au *oldAR* pour déclencher le *PrRtAdv*.

Pour initier un *Fast Handover*, le MN doit envoyer des *RtSolPr* au *oldAR* et ce dernier enverra un *PrRtAdv*. Dans le *RtSolPr*, le MN doit indiquer l'adresse MAC ou l'identification du point d'attache auquel le MN va être. Le *oldAR* va répondre avec un *Proxy Router Advertisement* qui contient les informations comme décrites dans la relève initiée par le réseau. Par la suite, le reste de la séquence de messages échangés est la même que pour la relève initiée par le réseau, avec les deux configurations possibles (*stateless* ou *statefull*) décrites auparavant. La *newCoA* est transmise au *newAR* pour validation en utilisant les messages HI et HAck.

2.4.2.3 Enregistrement

Les relèves initiées par le Mobile ou le Réseau ont le même processus d'enregistrement. Après le début de la relève, le MN envoie un FBU au *oldAR* en utilisant sa COA juste avant d'effectuer une relève. Le MN reçoit un FBBack du *oldAR* pour indiquer que la mise à jour est achevée. En effet, le *oldAR* envoie le FBBack au MN à travers le tunnel. Le *oldAR* peut aussi envoyer un FBBack au MN à travers son ancienne connexion (voir Figure 2.8). Si le HAck indique que la *newCOA* est valide, le *oldAR* s'apprêtera à expédier les paquets au MN à travers sa *newCOA*. Dans le cas contraire (pas de validation de l'adresse), le *oldAR* se préparera à envoyer les paquets au MN avec sa *newAR*.

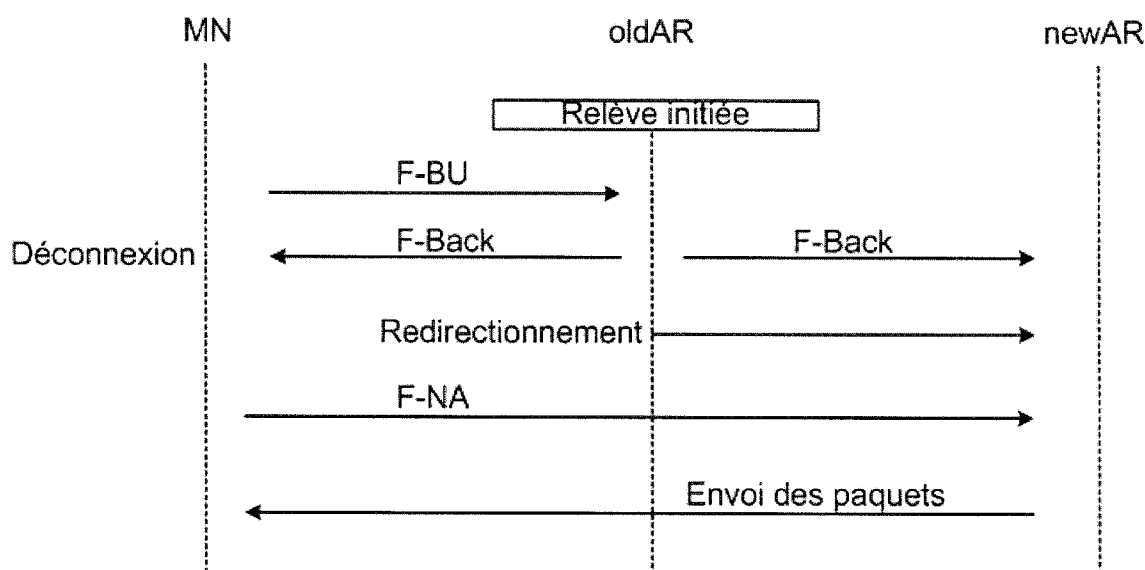


Figure 2.8 Messages échangés lors de l'enregistrement

Le *oldAR* n'arrêtera pas d'expédier les paquets pour le MN qu'une fois un FBU reçu par le MN montrant la fin de la relève. Ce FBU doit être envoyé au *oldAR*, pendant que le MN est toujours connecté au *oldAR*. Si ce n'est pas possible, le FBU ne doit être envoyé qu'une fois connecté au *newAR*. A ce moment là, le *oldAR* doit envoyer un FBack au MN localement ou à travers le *newAR*.

Après l'envoi du FBU et la réception du FBack, le *oldAR* peut commencer à expédier les paquets pour le MN à travers la *oldCOA* ou la *newCOA*. De plus, si la couche liaison supporte ce genre d'indications, le *oldAR* peut retarder le routage/expédition jusqu'à ce que le MN soit bien déconnecté du *oldAR*. Quand le MN arrive sur le *newAR*, il envoie un FNA pour initier la réception de paquets qui lui étaient destinés. Finalement après son déplacement, le MN doit envoyer des BU à son HA et ses CN à travers le *newAR*.

Lorsque le MN arrive sur le *newAR*, il envoie un message F-NA (*Fast Neighbor Advertisement*) pour initier le flot de paquets en attente. Finalement, après le déplacement du MN, le MN doit transmettre un BU à son HA et à ses CN à travers le *newAR* afin d'enregistrer sa *newCoA* (voir Figure 2.9).

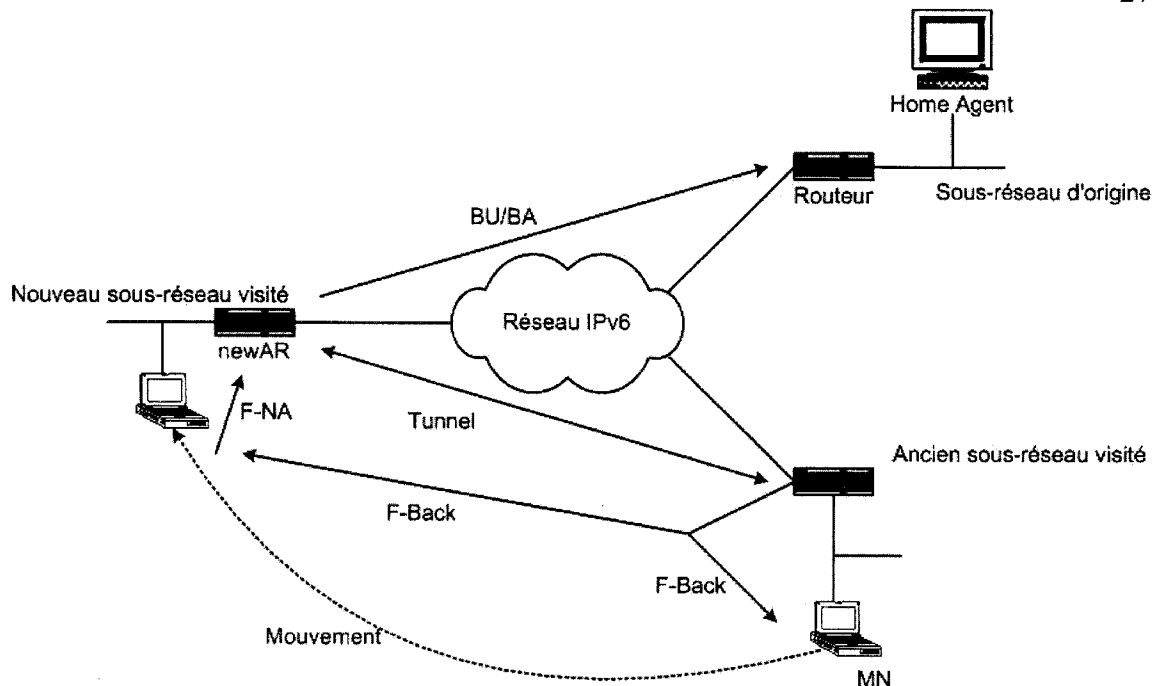


Figure 2.9 Messages échangés entre l'unité mobile et son HA, le oldAR et le newAR lors du déplacement du MN

2.4.3 Discussion

Le protocole FMIPv6, tel que défini par l'IETF, fonctionne en prenant en considération certaines hypothèses. Parmi ces hypothèses, l'unité mobile doit être capable d'anticiper la perte de la connexion avec un lien de manière à ce qu'il transmette un FBU avant d'être déconnecté du AR actuel. Si le mobile n'est pas capable d'envoyer le FBU avant de quitter le lien, le tunnel bidirectionnel entre l'ancien et le nouveau AR ne sera pas établi pendant que le Handover a lieu et les paquets qui seront envoyés à l'ancienne COA seront perdus.

Cependant d'après [9] et [10], le protocole peut être très sensible à certaines anomalies du réseau, et ne fonctionnera correctement que si toutes les hypothèses tiennent. Par exemple, une unité mobile doit être capable de déterminer d'avance le point d'accès où elle compte se rendre pour aboutir à une connectivité optimale. C'est un exemple de tâches qui ne peuvent pas être faites en temps normal. De plus, le document

ne spécifie que la communication entre les nœuds et pas les algorithmes utilisés pour prendre une décision de Handover, d'où il reste très difficile de faire une évaluation de l'efficacité du protocole sans se concentrer sur les propriétés de certaines implémentations particulières du protocole. Entre autre, le document de l'IETF [1] ne donne aucun conseil sur plusieurs problèmes critiques d'implémentation. Il est aussi possible ou même probable, que n'importe quelle implémentation particulière dans un environnement approprié soit complètement inadéquate dans un autre. Ainsi, n'importe quelle analyse précise du protocole demandera une grande concentration sur une analyse de certaines implémentations particulières.

2.5 Cellular IP

Cellular IP [8] est un nouveau protocole développé par l'IETF qui est optimisé pour permettre aux unités mobiles à grande mobilité d'accéder à Internet à travers Mobile IP. Cellular IP intègre un nombre important de concepts des réseaux cellulaires mais reste basé sur l'architecture IP tout en permettant des installations de taille pico à métropolitaine. La téléphonie mobile cellulaire est basée sur différents concepts de Mobile IP. Au lieu de cibler la mobilité globale, les systèmes cellulaires sont optimisés pour offrir des relèves rapides et efficaces dans des régions géographiques restreintes. Un protocole évolutif d'acheminement du trafic interconnecte les réseaux cellulaires entre eux pour permettre aux utilisateurs mobiles d'errer dans le réseau global.

2.5.1 Fonctionnement de Cellular IP

Cellular IP utilise les principes des systèmes cellulaires pour la gestion de la mobilité, la connexion passive et la gestion du handover, mais a une architecture basée sur le modèle IP. La composante principale du réseau *Cellular IP* est la station de base (*base station*) qui sert de point d'accès et en même temps de routeur des paquets IP, et intègre les fonctionnalités traditionnelles de contrôle cellulaire se trouvant dans le *Mobile Switching Center* (MSC) et la *Base Station Controllers* (BSC). Les stations de

base ressemblent à des systèmes d'acheminement de paquets IP normaux, seulement le routage IP est remplacé par le routage et la gestion de mobilité de *Cellular IP*. Le réseau *Cellular IP* est connecté à Internet à travers des *gateway routers*. La mobilité entre les *gateway routers* est gérée par Mobile IP alors que la mobilité dans les sous-réseaux est gérée par Cellular IP. Les unités mobiles utilisent l'adresse IP de la passerelle comme une COA. Lors de l'acheminement d'un paquet adressé au MN. Le paquet est d'abord routé au *Home Agent* du MN et ensuite expédié au *gateway router*. Le *gateway router* reçoit le paquet et l'expédie aux stations de base. Dans un réseau *Cellular IP*, les unités mobiles sont identifiées par leur *Home Address* (HAddr) et leurs paquets sont acheminés sans tunnel ni conversion d'adresses. Les paquets envoyés par le MN sont d'abord acheminés à la passerelle et ensuite sur Internet.

2.5.2 Handoff

L'algorithme *Hard Handoff* de Cellular IP est basé sur une approche simpliste de la gestion de mobilité qui permet une relève simple et rapide au prix d'une légère perte de paquets. La relève est initiée par l'unité mobile. Le terminal reçoit des signaux (*beacons*) des stations de base et commence une relève dépendamment de la force du signal reçu. Pour effectuer une relève, une unité mobile doit ajuster son interface radio sur la nouvelle station de base et envoyer un *Route Update*. Ce qui va mettre à jour le table de routage du *gateway* en configurant une nouvelle route pour joindre le MN. Le délai de la relève (*Handoff Latency*) est le temps entre la relève et le temps de réception du premier paquet à travers la nouvelle route. Pour la méthode *hard Handoff*, cela équivaut au temps d'envoi et de réception d'un paquet entre le MN et le *gateway*. Durant cette période, les paquets destinés au MN peuvent être perdus. La table de routage n'est pas effacée au moment de la relève, l'ancienne association expire après un moment. En effet, il existe un moment où la nouvelle et l'ancienne association sont valides et les paquets peuvent être délivrés aux deux stations de base. Cette fonction est utilisée lors du *Semisoft Handoff* de *Cellular IP* qui permet d'améliorer les performances de la relève

tout en restant une procédure non-pesante. *Semisoft handoff* rajoute un état additionnel aux états du mobile. Le *Semisoft Handoff* a deux composantes. Premièrement, pour réduire le délai de la relève, la nouvelle association pour le MN dans la table cache de routage doit être faite avant la relève. Quand le MN commence la relève, il envoie un paquet de type *semisoft* à la nouvelle station de base et se remet à l'écoute de l'ancienne station de base. Tout en restant en contact avec l'ancienne station de base, le paquet *semisoft* configure la table cache de routage. Après l'envoi du *semisoft*, le MN peut effectuer une relève normale. Le délai d'un *semisoft* peut être variable entre un aller-retour mobile-passerelle ou expiration de la route. Le délai permet de s'assurer que quand le MN ajuste sa radio sur la nouvelle station de base, ses paquets seront envoyés à travers l'ancienne et la nouvelle station de base.

2.5.3 Synthèse et Discussion

Tout d'abord, il est déjà dit que *Cellular IP* ne traitera pas de la mobilité globale et se concentrera plus sur la mobilité dans des régions géographiques restreintes. Cependant, cela ne fait toujours pas de Cellular IP un système qui peut procurer un *Seamless Handover* dans toutes les circonstances. En effet, une relève entre deux sous-réseaux de *Cellular IP* donnera lieu à la même perte de paquets que sous MIPv6 (sachant que la mobilité globale est gérée par MIPv6).

D'un autre côté, même si les paquets *semisoft* assurent au mobile de continuer à recevoir des paquets immédiatement après la relève, ils n'assurent pas une relève parfaite. Dépendamment, de la topologie du réseau et des conditions du trafic, le temps de transmission des paquets de la passerelle à l'ancienne et à la nouvelle station peut être différent et le flot de paquet envoyé à travers ces deux stations de base ne sera pas synchronisé au MN. Si la nouvelle station subit du retard, le MN recevra des paquets dupliqués. La réception de paquets dupliqués dans ce cas perturbera le fonctionnement des applications. Si, par contre, la nouvelle station de base subit de l'avance, à ce moment là les paquets seront déclarés perdus.

2.6 Simultaneous Bindings for Mobile IPv6 Fast Handovers

Fast Handover pour Mobile IPv6 (FMIPv6) est un protocole pour minimiser le temps d'interruption de la communication au cours d'un L3 Handover. *Simultaneous Bindings* est un protocole complémentaire au *Fast Handover* développé par l'IETF [9] pour minimiser le nombre de paquets perdus par le MN. Ainsi, le trafic pour le MN sera diffusé en *bicasting* ou *n-casting* à son emplacement actuel et aux autres endroits où le MN compte se rendre pour une courte période. Cette procédure permet d'éliminer l'ambiguïté sur le moment où le trafic sera redirectionné au nouvel emplacement du MN après un *Fast Handover* et permet aussi le découplage des *L2* et *L3 Handovers*. Il permet aussi de réduire le temps d'interruption du service en cas de ping-pong.

2.6.1 Terminologie

Bicasting/n-casting : Envoi du trafic destiné au MN simultanément à l'ancien AR et à un/plusieurs nouveaux AR.

Ping-pong : Mouvement rapide de va et vient entre deux points d'accès sans fil dû à un échec d'effectuer un *L2 Handover*. Ping-pong peut se produire si l'état radio de l'ancien/nouveau point d'accès sont les mêmes et ne permettent pas l'établissement d'une connexion (avec un taux d'erreur faible).

2.6.2 Fonctionnement du Simultaneous Bindings

Pour permettre une multitude de *simultaneous bindings*, le MN envoie une première demande pour enregistrer sa COA et envoie par la suite d'autres demandes d'enregistrer pour d'autres COA. Le récepteur de ces demandes d'enregistrement (ex. HA) gardera toutes ces adresses pour le MN au lieu de ne garder que la plus récente des adresses. Cela permettra d'effectuer un *bicasting* ou *n-casting* des paquets sur toutes les COA.

Simultaneous Bindings et *bicasting* peuvent être des outils intéressants pour découpler les *L3* et *L2 Handovers* et réduire la perte de paquets. Dans FMIPv6, ce mécanisme demande au récepteur du *Fast Binding Update* (FBU) avec l'option *Simultaneous Bindings* de faire plusieurs copies des paquets pour le MN et de les envoyer à toutes ses COA avant qu'il ne change d'emplacement. Ceci permet une meilleure relève minimisant/éliminant la perte de paquets. *Simultaneous Bindings* sont aussi utiles pour prévenir l'interruption de service lors d'un ping-pong.

2.6.2.1 Découplage du L3 et L2 Handovers

Le processus décrit dans FMIPv6 permet d'anticiper le *L3 Handover* de manière à ce que le trafic soit redirectionné au nouvel emplacement du MN avant d'y arriver. Cependant, ce n'est pas aussi simple de déterminer le bon moment pour commencer à expédier les paquets de l'ancien AR au nouveau AR, ce qui a un grand impact sur la rapidité et l'efficacité du Handover. Il se produira une perte de paquet si cela se passe trop tard ou trop tôt relativement au moment où le MN se détache du *oldAr* et se rattache au *newAR*. D'un autre côté, des mesures doivent être prises pour supporter le cas où le MN se déplace rapidement entre deux ARs (ping-pong).

Dans plusieurs réseaux sans fil, il est impossible de prédire exactement quand un MN va se détacher d'un lien sans fil du *oldAR* et s'attacher au *newAR*. Par conséquent, la prédiction du moment où les paquets doivent être acheminés entre le *oldAr* et le *newAR* est impossible. Certaines technologies sans fil utilisent des messages de la couche 2 qui informent le MN de faire une relève immédiatement ou simplement identifient que le MN s'est attaché/détaché. Même si les AR peuvent extraire l'information, il n'y aura pas suffisamment de temps pour que le *oldAR* détecte le détachement et commence à expédier les paquets au *newAR* avant que le MN soit rattaché au *newAR*. Ceci est dû au fait que le temps d'un L2 Handover est court (entre 10 s à 100 ms). Ainsi, une perte de la communication peut se produire à cause de cette incertitude, à moins que des améliorations soient apportées au processus de Handover.

Il est nécessaire de coupler le L2 et L3 Handover pour séparer le délai engendré par chacun d'eux. En effet, ils pourront se produire simultanément, ce qui diminuera le temps total de Handover. Une solution est de faire du *bicasting* ou *n-casting* pour une courte période. Cela veut dire que le processus du Handover décrit précédemment sera amélioré en ayant un envoi des paquets pour le MN par le *oldAR* à l'ancienne/nouvelle COA. Le MN sera donc capable de recevoir indépendamment de la durée exacte du L2 Handover.

2.6.2.2 Interruption de la connexion due au ping pong

Il est possible que le *L2 Handover* échoue ou se termine brusquement dans les systèmes sans fil. Par conséquent, un MN qui espère se déplacer entre le *oldAR* et le *newAR* peut ne jamais effectuer le *L2 Handover* et se trouver connecte au *oldAR* à nouveau. Un autre problème se produit quand le MN peut effectuer un mouvement de ping pong entre deux AR à cause de problèmes au niveau de la couche liaison. Dans ces deux cas, le MN est incapable de rétablir ses communications et doit transmettre un FBU dans FMIPv6 ou attendre l'établissement d'un tunnel bidirectionnel dans le cas assisté par la couche liaison avant de rétablir les communications.

Ceci peut être résolu en utilisant le *Simultaneous Bindings* qui permet au MN de maintenir la connexion au niveau 3 avec le *oldAR* pendant la durée du Handover, rendant ainsi le Handover plus efficace. Ceci élimine le besoin de transmettre continuellement des FBU dans FMIPv6 ou l'établissement du tunnel bidirectionnel.

2.7 Autres travaux effectués dans le domaine

Mis à part la multitude de travaux effectués par l'IETF pour fournir un service de routage et de gestion pour permettre aux unités mobiles d'errer à travers le réseau IPv6 sans perdre la connexion tout en étant localisable, beaucoup d'autres chercheurs se sont penchés sur le sujet et ont fourni d'autres approches intéressantes.

Dans [18], Vivaldi cherche à diminuer le délai et la perte de paquets lors de la relève en utilisant la méthode de *multicasting*. Hierarchical Mobile IPv6 (HMIPv6)

diminue le délai et la quantité de signalisation durant la relève. Cependant, HMIPv6 ne peut pas garantir les besoins d'un trafic sensible au délai ou en temps réel lors d'une relève inter-domaines MAP (cas de macro-mobilité). Ainsi, Vivaldi cible le problème du délai de la relève en présence d'une macro-mobilité. Le modèle proposé rétablit la connexion rapidement pour minimiser l'interruption des services qui se produit lors d'une relève. Il propose un traitement simultané de l'enregistrement de la nouvelle adresse dans le nouveau réseau et l'expédition des paquets au *newAR* pour le MN. Ceci peut être effectué en utilisant la méthode de *multicasting*. Le MAP multicast le paquet au *newAR* qui l'expédie au MN pendant la relève. Les résultats obtenus lors de la simulation montre une diminution de la perte de paquets et du délai durant une relève. Cependant, il reste que cette technique souffre d'un problème d'évolutivité (scalability). Vu que ce modèle utilise la méthode de *multicasting*, les ressources sont gaspillées à envoyer le même paquet plusieurs fois.

Dans [19], Chu et Weng essayent de traiter du problème de délai et de la perte de paquets lors de la relève en utilisant des pointeurs de renvoi dans le protocole MIPv6. En effet, dans des réseaux de grande envergure, le trafic de signalisation et le délai de la relève engendré par les enregistrements du protocole Mobile IP peuvent être importants si la distance entre le réseau visité et le réseau d'origine est grande. Le modèle proposé, Pointer Forwarding MIPv6 Mobility Management, utilise une chaîne de pointeurs dans un réseau Mobile IP. Cette approche diminue le coût des mises à jour des *Home Agents* et des *Correspondent Nodes*. En effet, le réseau est réparti en domaines appelés *Mobility Domains* (MD). Les MD sont gérés par des Pointer Forwarding Mobility Agent (PFMA) qui ont pour rôle de connaître la localisation du MN dans leur domaine ou dans tout autre domaine où il se trouve. Lorsqu'un MN change de MD, il s'enregistre auprès du PFMA du nouveau MD et demande à l'ancien PFMA de créer un pointeur de renvoi vers le nouveau PFMA. Ainsi, lorsqu'un *Correspondent Node* envoie un paquet au MN, il est d'abord acheminé au premier PFMA qui l'expédie au prochain PFMA de la chaîne de

pointeur jusqu'à réception par le MN. Lorsque la chaîne atteint une taille maximale de N pointeurs, un pointeur est créé au PFMA d'origine vers le dernier PFMA de la chaîne.

Ce modèle est très efficace. Cependant, HMIPv6 devance ce modèle vu qu'à chaque fois qu'une recherche du MN est faite, cela peut prendre N expéditions de PFMA en PFMA avant de trouver le MN. D'un autre côté, dans HMIPv6, un MN peut se déplacer dans plus de N cellules dans le même domaine MAP sans mettre à jour son HA.

Dans [20], Omae propose un modèle pour améliorer les performances de la relève sous Mobile IPv6 en faisant des modifications sur l'unité mobile. En effet, Mobile IP ne peut pas effectuer une relève sans coupure de la connexion du délai de la signalisation entre le HA et le MN. Durant cette période, les paquets destinés au MN sont perdus. L'introduction de HMIPv6 minimise la période d'interruption du service puisque cela prend moins de temps pour mettre à jour le MAP qu'un HA distant. Dans ce modèle, Omae propose que les paquets UDP sortant soit sauvegardés pour une transmission ultérieure lorsque la connexion sera rétablie. Les paquets TCP entrant seront retransmis puisque les accusés de réception ne pourront pas être envoyés par le MN. En ce qui concerne les paquets TCP sortant, ils seront sauvegardés pour une transmission ultérieure. Ce modèle donne des résultats globaux satisfaisants mais reste que les paquets UDP entrants sont tous perdus et que les paquets TCP entrants seront retransmis après un délai, ce qui cause un gros problème pour le trafic temps réel et le trafic sensible au délai. D'un autre côté, les paquets UDP ou TCP sortant sont toujours envoyés après que la connexion eut été rétablie, ce qui cause un retard dans la communication.

Dans [21], Patanapongpibul et Mapp proposent un modèle capable de supporter des relèves dans les réseaux IPv6 sans introduire un nouveau protocole de gestion de la mobilité ou changer l'infrastructure existante. En effet, ce modèle permet à l'unité mobile de décider du moment de la relève. Ainsi, les modifications seront faites sur l'unité mobile. Le mécanisme implémenté dans l'unité mobile est censé contrôler et

forcer les relèves, choisir le meilleur lien pour effectuer la relève et faire une relève au moment approprié. En effet, l'article détaille la manière de forcer la relève et la manière de choisir le meilleur lien pour la relève mais ne donne aucun détail sur la manière dont la relève sans coupure se fait.

2.8 Patron de mobilité

La mobilité sans fil a beaucoup été étudiée ces dernières années. Cependant, la plupart des études récentes se concentrent sur les protocoles de la couche réseau et seulement quelques-uns d'entre eux s'intéressent à l'aspect de la mobilité engendré par le comportement des usagers. D'après [15], pour créer des applications mobiles intelligentes avec une gestion du routage efficace, il est important qu'un système soit plus intelligent en aidant ses utilisateurs à se déplacer. La mobilité est un aspect du comportement des usagers qui affecte les réseaux sans fil. Pour maintenir un service de haute qualité sans interruption dans un environnement mobile, le routage des informations ne doit pas être fait seulement de manière passive mais aussi de manière agressive en anticipant le comportement des usagers mobiles. Ainsi, un ensemble d'algorithmes de prédiction du déplacement des mobiles [15] (*Mobile Motion Prediction* ou MMP) peuvent être utilisés pour prédire l'emplacement futur de l'utilisateur. L'utilisateur pourra donc immédiatement recevoir ses services ou paquets avec la même efficacité qu'avant son déplacement. Ces algorithmes sont basés sur le fait que tout le monde a un certain degré de régularité dans ses déplacements. En effet, le déplacement des personnes consiste de déplacements aléatoires et d'autres réguliers, et la majorité des usagers mobiles ont des régularités journalières (par heure, hebdomadaire ...) et suivent ces patrons plus ou moins chaque jour de la semaine (à des heures précises, des jours précis ...).

2.8.1 Synthèse des méthodes de prédiction du déplacement des usagers

Plusieurs modèles de mobilité d'utilisateurs et de déplacement peuvent être trouvés dans la littérature pour représenter la mobilité des utilisateurs. Parmi eux, un modèle très utilisé est le modèle de mouvement fluide (*Fluid Flow*) [15] dans lequel il est assumé que les utilisateurs mobiles sont répartis uniformément et qu'ils se déplacent à une vitesse moyenne avec une direction de mouvement de $[0, 2\pi]$. Avec les suppositions précédentes, ce modèle a été efficacement utilisé pour étudier la performance des réseaux PCS (*Personal Communications System*) et l'impact de la mobilité des utilisateurs sur leurs bases de données. Pour étudier la relève dans les systèmes mobiles, un modèle de mobilité [15] a été proposé qui assume que la vitesse et la direction d'un terminal mobile sont des variables aléatoires indépendantes. La direction des mobiles est supposée être uniformément distribuée dans $[-\pi/2$ et $\pi/2]$. Il est aussi supposé que lorsque les mobiles traversent les frontières des cellules, leurs vitesses et directions sont régénérées aléatoirement. Ces modèles sont utiles et efficaces pour analyser la performance des systèmes mobiles d'un point de vue « macro » (pour se concentrer sur la population globale des utilisateurs mobiles). Cependant, ils ne sont pas appropriés pour prévoir le déplacement des utilisateurs mobiles puisque les caractéristiques de chaque utilisateur mobile ne sont pas considérées séparément.

Les méthodes de prédiction du déplacement des utilisateurs ont beaucoup été utilisées dans les systèmes mobiles sans fil, voilà deux cas d'utilisation :

- Stratégie Gestion de la Mobilité avec profil d'utilisateur

Les systèmes PCS sont caractérisés par une grande densité et grande mobilité d'utilisateurs. On s'attend à ce qu'il y ait beaucoup de trafic généré à cause du nombre important d'enregistrements effectués au niveau radio. Un système utilisant le profil d'utilisateur est proposé dans [13] pour réduire la signalisation au niveau radio en augmentant l'intelligence dans le réseau cœur. Les résultats de ce travail montrent qu'à travers un grand nombre de paramètres, il est possible de réduire la charge de signalisation au niveau radio et du réseau cœur pour une augmentation modeste du délai d'établissement de la communication.

- Protocole pour Mobile MPLS avec profil d'utilisateur

Mobile MPLS est un nouveau modèle qui intègre Mobile IP et MPLS, pour que le protocole MPLS puisse supporter la mobilité. Dans Mobile MPLS [14], quand un CN veut communiquer avec un usager mobile, le CN doit d'abord envoyer ses paquets au HA qui les expédiera au MN à travers *Label Switched Path* (LSP). Cette méthode d'expédition triangulaire de paquets est inévitable même si un routage optimisé est utilisé par la suite. Cependant, si le comportement de l'utilisateur est connu avant l'expédition des paquets, cet acheminement triangulaire peut être évité.

CHAPITRE III

ALGORITHMES ET MODÈLE DE GESTION DE MOBILITÉ

Dans ce chapitre, nous proposons un nouveau modèle de gestion de mobilité pour les réseaux IPv6 se basant sur les patrons de mobilité des usagers, appelé *Seamless Mobile IPv6*. Ce modèle vise à réduire les coûts de la signalisation engendrée par la mise à jour des correspondants, ainsi que les délais de relèvement entre les réseaux lors des déplacements de l'unité mobile. Nous présentons d'abord le principe de base de ce modèle. Ensuite, nous introduisons le module d'apprentissage des patrons de mobilité, pour continuer avec notre modèle de gestion de mobilité et ses différents algorithmes. Finalement, nous modélisons les coûts de signalisation et les délais de relèvement comparativement à d'autres stratégies de gestion de mobilité.

3.1 Principe de base

Le modèle proposé est basé sur le protocole de gestion de mobilité Mobile IPv6 [1]. En effet, notre modèle utilisera la même architecture et les mêmes entités utilisées dans MIPv6 (voir section 2.1). *Seamless Mobile IPv6* (SMIPv6) tâche principalement de réduire le délai de la relève afin d'approcher au maximum une relève sans coupure et de minimiser la signalisation. Dans le modèle Mobile IPv6 [1], l'unité mobile met à jour ses correspondants et son HA de sa nouvelle adresse une fois qu'il franchit la frontière d'un nouveau réseau. Le délai engendré par la détection du mouvement, la création d'adresse, la détection de l'unicité de cette adresse et la mise à jour des correspondants fait en sorte que l'unité n'est pas joignable pendant un intervalle de temps important. De plus, une unité mobile qui ne se déplace pas est obligée de générer une signalisation importante pour mettre à jour ses correspondants afin de les aviser de son adresse qui est inchangée.

SMIPv6 se base sur les patrons de mobilité des usagers afin de prédire la prochaine cellule où la relève va se produire afin de mettre à jour ses correspondants avant de quitter son réseau courant. D'un autre côté, SMIPv6 peut prédire le moment de la relève grâce à des informations provenant de la couche 2. De plus, SMIPv6 utilise une méthode de création d'adresses IPv6 uniques qui permet d'éliminer la vérification de l'unicité des adresses appelée *Duplicate Address Detection* (DAD) [22]. Ainsi, la prédiction du moment et du lieu de la relève, combinés avec la création d'adresses IPv6 uniques à l'avance, permet d'approcher une relève sans coupure. L'usage du patron de mobilité va permettre d'envoyer un message de mise à jour à ses correspondants dès qu'un changement de réseau est en vue, au lieu d'envoyer une multitude de messages à intervalle défini.

Le modèle de gestion de mobilité SMIPv6 se compose de deux entités : un module d'apprentissage des patrons de mobilité et un protocole de gestion de la mobilité. Le module d'apprentissage sera implémenté dans les unités mobiles, alors que le protocole de gestion de mobilité sera implémenté dans toutes les entités. Notre travail consistera seulement à développer le modèle de gestion de mobilité. Cependant, notre modèle de gestion de mobilité se base sur le module d'apprentissage des patrons de mobilité. Ainsi, nous allons présenter les spécifications du module d'apprentissage ainsi que nos attentes de ce module mais nous ne détaillerons pas le fonctionnement de ce dernier.

3.2 Module d'apprentissage des patrons de mobilité

Dans la vie courante, la majorité des usagers mobiles ont des régularités journalières (par heure, hebdomadaire ...) et suivent ces patrons plus ou moins chaque jour de la semaine (à certaines heures, à certains jours...). En effet, ces usagers vont quitter la maison pour aller au travail, quitter le travail pour déjeuner à midi, retourner à la maison le soir, aller à la salle de sport, etc. Si ce patron de mobilité pouvait être déterminé de manière précise, la signalisation pour les mises à jour sera réduite ainsi que

le délai de la relève. Dans cette section, nous parlerons davantage de patron de mobilité des usagers ainsi que du modèle d'apprentissage des patrons.

3.2.1 Représentation du réseau

Nous considérons un réseau cellulaire qui couvre une grande zone géographique. Ce réseau se décompose en cellules ayant une forme hexagonale [23] et chaque cellule contient une station de base. La Figure 3.1 illustre la représentation de notre réseau. Chaque cellule est numérotée à l'aide d'un couple (x, y) qui représente le numéro de la ligne et le numéro de la colonne. Ainsi, la numérotation reflète la proximité entre les cellules. Un déplacement de l'unité mobile entre deux cellules $C1(x1, y1)$ et $C2(x2, y2)$ se caractérise par un couple (dx, dy) qui représente une différence de position : $dx = x2 - x1$ et $dy = y2 - y1$. Quand l'unité mobile sort d'une cellule, elle doit entrer dans une cellule voisine. Avec des déplacements de ce genre, l'unité mobile se déplace avec des petites valeurs entières de dx et dy étant -1, 0 ou 1. D'après [23], l'utilisation d'une numérotation non-arbitraire permet une certaine cohérence dans les déplacements générés.

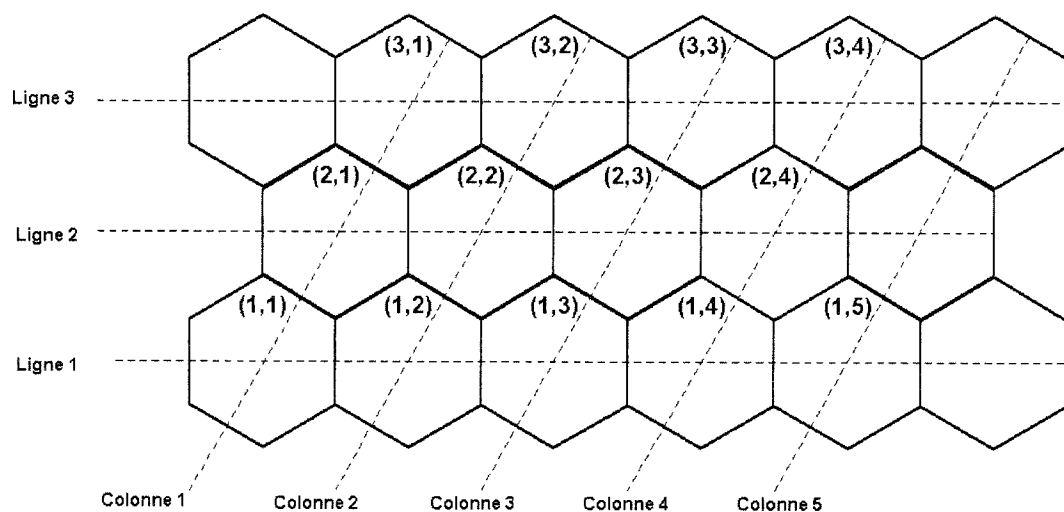


Figure 3.1 Représentation du réseau cellulaire

3.2.2 Patron de mobilité des usagers

Pour comprendre la représentation du déplacement des unités mobiles afin de déterminer les patrons de mobilité, nous allons examiner un comportement typique d'un usager mobile. La Figure 3.2 illustre le déplacement de notre usager. La cellule d'origine de l'unité mobile est la cellule (1,1), représentée par la lettre *O*, et son travail se trouve à la cellule (1,5), représenté par la lettre *T*. Supposons que chaque jour à 9:00 am l'utilisateur quitte son domicile et va à son travail à travers les cellules (2,1), (3,1), (3,2), (3,3), (2,4), (1,5). A 12:00 pm, il va déjeuner au restaurant à la cellule (1,4). L'après-midi, vers 6 :00 pm, il rentre chez lui en faisant le chemin inverse. Il va quelques fois à la salle de sport se trouvant à (2,2), et dîne souvent dans un restaurant à la cellule (1,2). Nous avons besoin d'un modèle pour retrouver ce patron de mobilité. Cependant, un patron de mobilité irrégulier et complexe comme celui d'un représentant commercial serait difficile à représenter et nous utiliserons le modèle de gestion de mobilité par défaut.

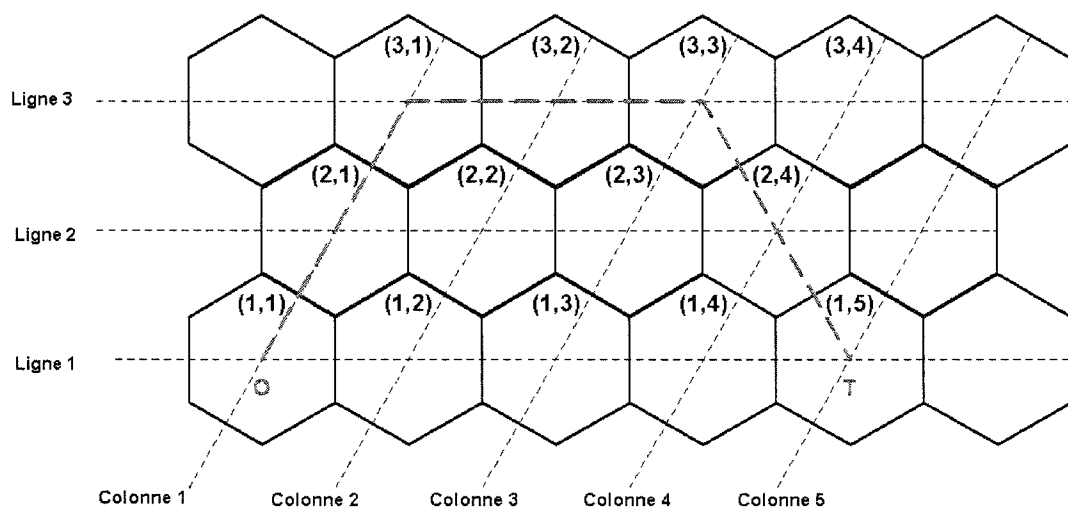


Figure 3.2 Exemple d'un patron de mobilité

Nous observons que la direction du mouvement dépend de la position actuelle (x, y) et du temps t . Quand l'utilisateur se trouve dans la cellule $(2,1)$ entre 9:00 am et 10:00 am, nous pouvons prédire qu'il se déplace vers la cellule $(3,1)$. Ainsi, pour chaque position et moment de la journée, il existe un patron de mobilité plus ou moins stable. Il y aura aussi quelques déplacements temporaires ou occasionnels qui n'ont pas été prévus dans le patron de mobilité. Le modèle devra être capable d'apprendre à ignorer les déplacements temporaires imprévus.

3.2.3 Modèle d'apprentissage du patron de mobilité

Nous considérons un patron de mobilité qui se répète sur une durée de 7 jours. En effet, la mobilité d'un usager sera différente dépendamment si nous sommes un jour de semaine ou de fin de semaine. Nous considérons un patron de mobilité d'un jour de semaine sur une topologie hexagonale bidimensionnelle. Nous supposons que chaque usager mobile a un identifiant unique généré avec la méthode 64-bit *Extended Unique Identifier* (EUI-64 par IEEE) et que chaque station de base diffuse un préfixe de sous-réseau que les unités mobiles dans cette cellule peuvent recevoir à travers les *Router Advertisements* (RA). Ainsi, à chaque moment, l'unité mobile connaît le sous-réseau où il se trouve.

L'unité mobile ramassera les informations concernant sa mobilité quotidienne et le système apprendra à connaître le patron de mobilité de cet usager grâce à un des algorithmes d'apprentissage de la mobilité de la catégorie *Mobile Motion Prediction Algorithms* (MMP) [15]. Le modèle générera un premier patron de mobilité et, au fur et à mesure des déplacements, il vérifiera si l'unité mobile suit son patron. Sinon, le modèle corrigera le patron jusqu'à ce que celui-ci reflète les déplacements de l'unité mobile. Une fois que l'apprentissage converge et que les tests confirment que le patron de mobilité a été appris, l'unité mobile sera capable d'utiliser son patron de mobilité pour prédire son emplacement futur. Les propriétés importantes que le système d'apprentissage doit avoir sont :

1. il doit être capable d'apprendre les déplacements répétitifs, s'ils existent;
2. dépendamment du moment de la journée, le système doit être capable de donner un ensemble de cellules dans lesquelles l'utilisateur devrait se trouver;
3. le système doit être capable de détecter et réapprendre, si le mobile change ses habitudes de déplacement.

Le modèle d'apprentissage a trois entrées : les coordonnées x et y de la position actuelle, et le temps actuel t . L'unité d'apprentissage prend ces informations et donne, selon la position actuelle et le temps, la direction (dx , dy) que l'unité mobile devrait prendre. Un modèle d'apprentissage est illustré à la Figure 3.3.

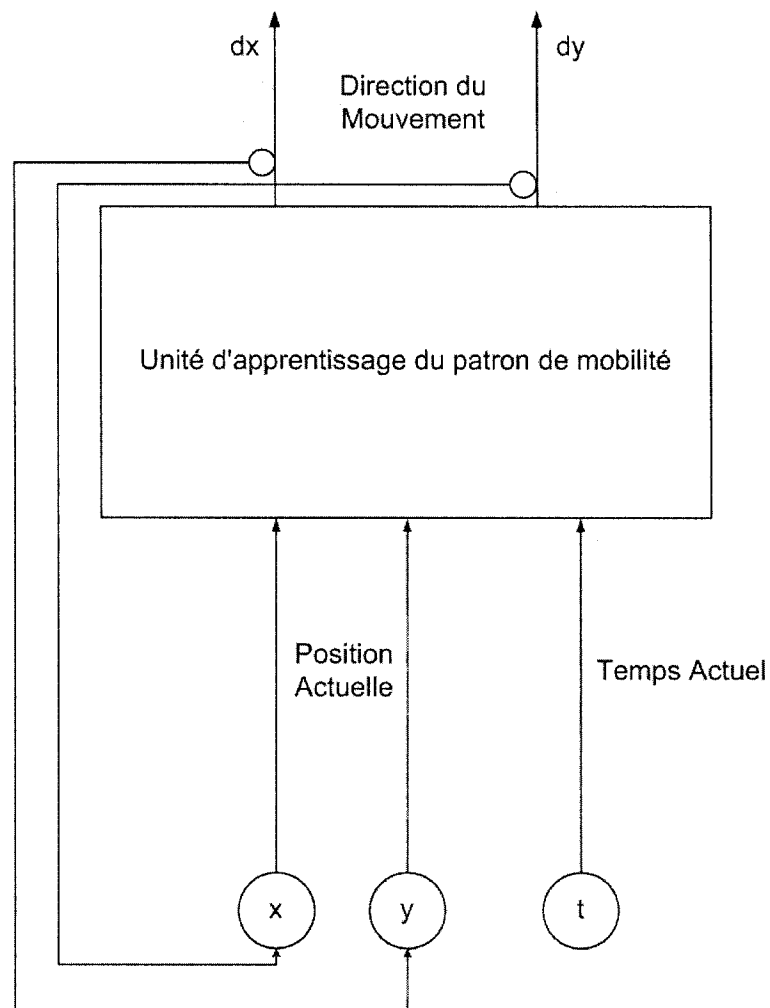


Figure 3.3 Modèle d'apprentissage des patrons de mobilité

En réalité, le modèle devrait être un simple programme. Durant la période d'apprentissage, les deux connecteurs de rétroaction reliant les sorties aux entrées seront déconnectés. Ainsi, les informations fournies au système seront : la position et le temps actuels. Quand l'utilisateur ne se déplace pas, les deux sorties indiquent $\langle 0,0 \rangle$. Ce système d'apprentissage devrait produire un historique des déplacements, comme le montre le Tableau 3.1.

Tableau 3.1 Exemple d'un historique des déplacements

Colonne	Heure	Jour 1		Jour 2	
		Position	Déplacement	Position	Déplacement
01	0:00	(3,4)	(0,0)	(2,5)	(+1,-1)
02	0:01	(3,4)	(0,0)	(3,4)	(0,0)
03	0:02	(3,4)	(0,0)	(3,4)	(0,0)
04	0:03	(3,4)	(0,0)	(3,4)	(0,0)
05	0:04	(3,4)	(0,0)	(3,4)	(0,0)
06	0:05	(3,4)	(+1,+1)	(3,4)	(0,0)
07	0:06	(4,5)	(0,0)	(3,4)	(0,0)
08	0:07	(4,5)	(+1,0)	(3,4)	(+1,0)
09	0:08	(5,5)	(-1,-1)	(4,4)	(0,0)
10	0:09	(4,4)	(0,0)	(4,4)	(0,0)
11	0:10	(4,4)	(0,0)	(4,4)	(0,0)
12	0:11	(4,4)	(0,-1)	(4,4)	(0,0)
13	0:12	(4,3)	(0,0)	(4,4)	(0,0)

Il y a plusieurs modèles, de différentes complexités, capables d'implémenter ce module d'apprentissage. Nous ne donnerons pas beaucoup de détails sur le module d'apprentissage vu que nous ne l'implémentons pas. D'après [15], un choix judicieux serait l'utilisation d'un réseau de neurones (*feed-forward neural network*). En effet, les réseaux de neurones sont des modèles capables d'apprendre, de généraliser ou d'organiser les données en utilisant un traitement parallèle. Parmi tous les genres de réseaux de neurones, les plus utilisés sont les réseaux de neurones *feed-forward* à multicouches. Ces réseaux sont capables de trouver des relations non-linéaires entre les

entrées et sorties et ont été acclamés comme des «Approximateurs Universels ». Le réseau aurait trois entrées : deux représentant la position actuelle de l'unité mobile et une entrée représentant le temps actuel. Il aura aussi deux sorties indiquant le déplacement de l'unité mobile.

3.3 Modèle de gestion de mobilité

Ce modèle se base sur les patrons de mobilité des usagers générés par le module d'apprentissage. En effet, un usager commencera par entraîner son module d'apprentissage jusqu'à ce qu'il détermine un patron de mobilité. Ce patron de mobilité sera enregistré dans la mémoire de l'unité mobile. À ce moment là, l'unité mobile est capable d'utiliser son patron de mobilité pour envoyer à ses correspondants un message, appelé *Binding Update* (BU), comportant son emplacement futur selon l'heure et sa position actuelle. L'unité mobile devra envoyer ce message avant d'entrer dans le nouveau réseau. Selon les caractéristiques du réseau, le modèle de gestion de mobilité utilisera des informations communiquées par la couche 2 sur la présence d'un nouveau point d'accès et la possibilité de relève. Nous ne pouvons pas utiliser les informations temporelles dans le patron de mobilité pour prédire l'instant exact (à la milliseconde près) où aura lieu la relève. Pour que la relève se produise sans coupure, le message envoyé aux autres usagers pour les prévenir d'un futur changement d'adresse dû à une relève, comportera l'adresse de l'unité mobile dans le réseau prochainement visité. Cette adresse sera pré-configurée connaissant le prochain réseau visité et son préfixe de sous réseau. Elle sera toujours la même dans le même sous-réseau vu que les adresses sont toujours construites de la même manière, en utilisant les mêmes informations. Nous commencerons par détailler l'architecture du modèle de gestion de mobilité. Nous introduirons ensuite les concepts de profil de mobilité et de prédiction du moment de relève avant de passer en revue les algorithmes qui régissent notre modèle.

3.3.1 Architecture du modèle de gestion de mobilité

Il y a deux architectures possibles pour que les unités mobiles (MN) puissent mettre à jour leurs correspondants en leur envoyant un BU comprenant leur futur emplacement provenant de leur patron de mobilité. La première est une architecture répartie et la deuxième une architecture centralisée. La Figure 3.4 illustre l'architecture répartie qui permet aux MN de mettre à jour leur position auprès de leurs correspondants (CN). Dans cette approche, chaque utilisateur a un module d'apprentissage dans son dispositif mobile qui génère son patron de mobilité. Ce patron de mobilité est gardé dans la mémoire de ce dispositif mobile. Ainsi, quand un correspondant devra être mis à jour, un BU généré à partir de ce patron lui sera directement envoyé.

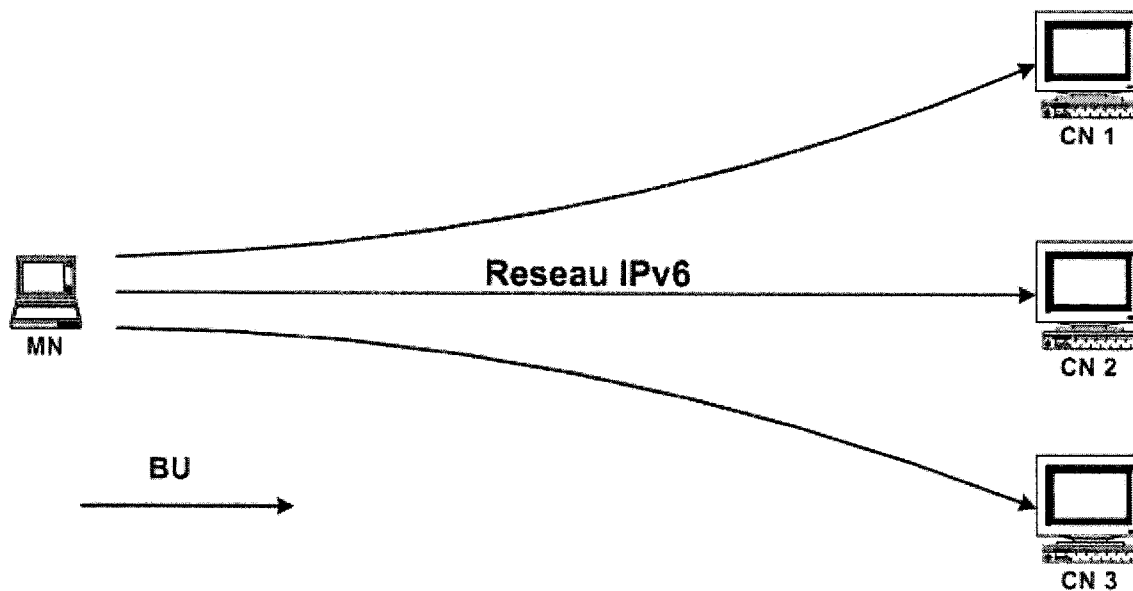


Figure 3.4 Architecture répartie

L'approche centralisée aura une base de données centrale où tous les patrons seront stockés. La Figure 3.5 illustre cette méthode centralisée d'envoyer des BU. Quand l'architecture centralisée est utilisée, chaque utilisateur a un module d'apprentissage dans son dispositif mobile qui génère son patron de mobilité. Ce patron de mobilité sera enregistré dans la base de données du serveur de patrons de mobilité et l'unité mobile pourra mettre à jour son patron s'il y a des changements. Le serveur de patrons de mobilité va se charger de mettre à jour les correspondants des unités mobiles dont il se

charge, en leur envoyant des BU comportant l'emplacement futur de l'unité mobile en question. Après un certain temps, le patron de mobilité expirera et sera supprimé de la base de données. Si le serveur tombe en panne, toutes les données seront perdues. Malgré l'utilisation d'un système de sauvegarde, nous allons faire face à un problème de synchronisation.

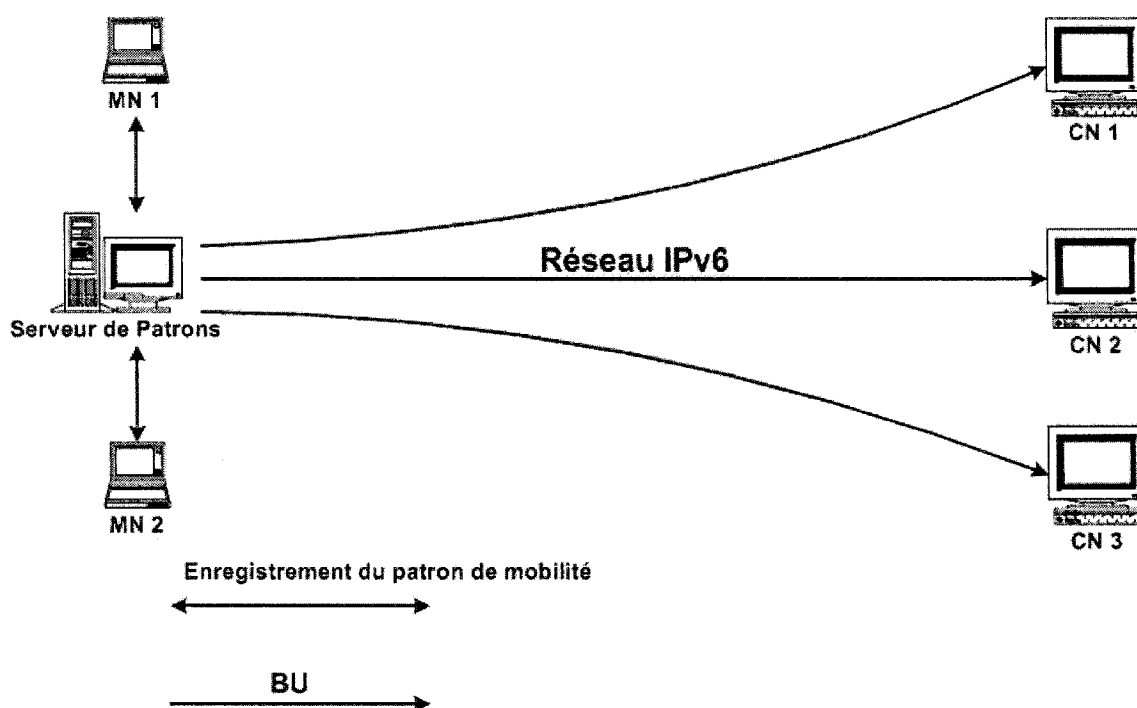


Figure 3.5 Architecture centralisée

3.3.2 Profil de mobilité des usagers

Dans cette section, nous faisons abstraction du module d'apprentissage. Notre unité mobile peut avoir une multitude de profils de mobilité qui constituent son patron de mobilité global. En effet, une fois l'historique des déplacements analysé, nous allons retrouver plusieurs profils de mobilité qui vont englober tous les déplacements majeurs de l'utilisateur. D'après [11], un usager régulier devrait avoir environ six profils différents. Les profils principaux que nous pourrions retrouver, pourraient être des profils pour aller au travail, revenir du travail, aller déjeuner au restaurant, aller à la salle de sport, aller

faire les courses, etc. Un profil couvre un déplacement qui se produit régulièrement et prend la même durée pour être effectué. Le profil va comporter des informations sur les cellules visitées durant ce déplacement, les réseaux auxquels elles appartiennent ainsi que les temps d'entrée et de sortie de ces réseaux. La Tableau 3.2 illustre le profil pour se rendre au travail de l'exemple 3.2.2.

Tableau 3.2 Exemple de profil de mobilité

Cellule	Préfixe réseau	te entrée	ts sortie
(1,1)	AC-DE-48-23-45-67-AB-11	8:00	8:10
(2,1)	AC-DE-48-23-45-67-AB-21	8:10	8:15
(3,1)	AC-DE-48-23-45-67-AB-31	8:15	8:22
(3,2)	AC-DE-48-23-45-67-AB-32	8:22	8:35
(3,3)	AC-DE-48-23-45-67-AB-33	8:35	8:40
(2,4)	AC-DE-48-23-45-67-AB-24	8:40	8:50
(1,5)	AC-DE-48-23-45-67-AB-15	8:50	9:00

Le Tableau 3.2 représente le déplacement journalier d'un usager mobile pour se rendre à son travail. L'usager sort de chez lui (1,1) à 8:00 am et arrive au travail (1,5) à 9:00 am. Il entre dans les cellules (2,1), (3,1), (3,2), (3,3), (2,4) durant son trajet à 8:10 am, 8:15 am, 8:22 am, 8:35 am, 8:40 am, 8:50 am approximativement.

En effet, un MN suit son profil approximativement dans le sens où il fera le même parcours que dans le profil de mobilité, mais ne sera pas aussi précis sur le temps (à la seconde ou à la minute) de changement de cellules. Le MN peut toujours aller au travail en faisant exactement le même parcours mais il sort de chez lui entre 7:55 am et 8:05 am. Selon le trafic, le MN franchira les cellules avec un peu de retard ou d'avance par rapport au temps indiqué dans le profil de mobilité. Nous introduisons donc un facteur retard (*delay*) qui servira à absorber cette irrégularité.

Au moment où une unité mobile devra mettre à jour ses correspondants, elle leur enverra un BU avec sa prochaine adresse avec une durée de vie correspondant au temps de sortie de cette cellule plus l'estimation du retard ($bu_lifetime + delay$). Le fait

d'envoyer une seule adresse à la fois permettra de minimiser la congestion du réseau avec un excédant d'informations car, si un correspondant à besoin d'une seule adresse pour envoyer un paquet au MN, il serait inutile de lui envoyer tout le profil de mobilité. D'un autre côté, pour la protection des informations des usagers, nous ne pouvons pas divulguer le patron de mobilité d'un usager aux autres utilisateurs.

3.3.3 Prédiction du moment de la relève

Pour prédire le moment où aura lieu une relève, nous utiliserons des indications provenant de la couche 2 indiquant l'approche d'un nouveau point d'accès. Cette méthode, semblable à celle utilisée dans *Fast Handovers for Mobile IPv6* (FMIPv6) [6], consiste à anticiper la relève par l'usage d'informations provenant de la couche 2, comme la réception d'un meilleur signal d'autres points d'accès suivi de la dégradation du signal du point d'accès actuel. Ce genre d'événements provenant de la couche 2 oblige l'unité mobile à commencer une relève à la couche 3 (*L3 Handover*) si le point d'accès n'appartient pas au même sous-réseau.

En effet, pour prédire une relève, l'unité mobile doit être capable de détecter le mouvement de l'unité mobile comme la pénétration d'un nouveau sous-réseau. La représentation du réseau utilisée auparavant (voir Figure 3.1), étant l'utilisation de cellules à forme hexagonale, ne reflète pas la réalité. Dans la vraie vie, les cellules ont des zones de couvertures rondes ayant un rayon défini et, pour assurer une bonne couverture, les cellules ont des zones de chevauchement entre elles où l'unité mobile peut recevoir le signal des deux points d'accès voisins.

Une unité mobile qui se déplace d'une cellule à une autre va pénétrer cette zone de chevauchement et effectuer une relève entre les deux points d'accès à la couche 2 sans coupure. Cependant, une unité mobile peut pénétrer cette zone de chevauchement et ne pas se rendre compte qu'elle a pénétré un nouveau sous-réseau. En effet, l'information concernant le préfixe de sous-réseau provient de la couche 3 dans les *Router Advertisements* (RA) diffusés par les routeurs d'accès. Ces RA sont diffusés dans

tout le sous-réseau à un intervalle défini *RtrAdvInterval* qui est relativement important (pour ne pas saturer le réseau avec la signalisation). Donc, une unité mobile peut recevoir le signal d'un nouveau point d'accès et ne pas encore recevoir de RA. À ce moment là, l'unité mobile ne se rendra compte qu'elle a pénétré un nouveau sous-réseau qu'après un délai allant à *RtrAdvInterval* où elle recevra son premier RA.

Notre modèle de gestion de mobilité utilise les patrons de mobilité pour prédire le prochain sous-réseau à pénétrer, donc nous pouvons enclencher une relève à la couche 3 sans attendre le RA. En effet, le patron de mobilité contient l'information sur les cellules et le sous-réseau auquel elles appartiennent (voir Tableau 3.2). Donc, une fois le signal d'un nouveau point d'accès reçu, nous pouvons dire si le MN va subir une relève à la couche 3.

3.3.4 Formation des adresses IPv6

Une unité mobile qui se déplace tout en prenant en compte son patron de mobilité pour prédire le moment où aura lieu une relève entre deux réseaux IPv6, cherche à envoyer à ses correspondants un message de mise à jour (BU) de son adresse dans un réseau prochainement visité. En effet, l'unité mobile va former une adresse IPv6 valable dans le prochain réseau et ceci, en se basant sur le préfixe du réseau recueilli dans le profil de mobilité. Les unités mobiles utiliseront la méthode *IPv6 Stateless Address Autoconfiguration* [22]. Cette méthode ne requière aucune configuration manuelle des unités mobiles, ni de serveur additionnel, mais seulement une configuration minimale des routeurs. La méthode *Stateless* permet à l'unité de générer sa propre adresse en utilisant une combinaison d'informations locales et d'informations publiées par les routeurs d'accès à travers les *Router Advertisements*. Les routeurs d'accès diffusent des préfixes qui identifient le sous-réseau associé au lien, alors que les unités mobiles génèrent un identifiant qui identifie l'interface entre le réseau et cette unité. Une adresse est formée en combinant les deux.

Dans le cas des réseaux IPv6, les adresses sont de 128 bits. Ainsi, selon [24], il est nécessaire que l'identifiant d'interface soit de 64 bits et construit selon le format IEEE EUI-64 [24]. Les identifiants EUI-64 sont formés par la combinaison d'un identifiant unique 24 bits de la compagnie enregistré par IEEE *Registration Authority* et une extension de 40 bits attribuée par la compagnie qui devrait être unique. La méthode de création ou de dérivation d'identifiants d'interface 64-bit *Extended Unique Identifier* à partir d'adresse MAC-48 et EUI-48 a pour but de maintenir une unicité globale des identifiants. Cependant, il n'y a aucune garantie de duplication due aux erreurs. La Figure 3.8 illustre la structure d'un identifiant EUI-64.

Identifiant de la compagnie			Extension					
eui[0]	eui[1]	eui[2]	eui[3]	eui[4]	eui[5]	eui[6]	eui[7]	ordre
AC	DE	48	23	45	67	AB	CD	hex

Figure 3.6 Structure d'un identifiant EUI-64

Vu que nous connaissons déjà le préfixe du réseau grâce aux informations dans le profil de mobilité, nous pouvons créer l'adresse IPv6 de l'unité mobile dans le prochain réseau en combinant le préfixe du réseau avec l'identifiant unique EUI-64. Ainsi, l'adresse IPv6 sera unique dans ce réseau vu qu'elle est basée sur un identifiant unique et nous pouvons éviter de vérifier l'unicité de l'adresse en faisant un *Duplicate Address Detection* (DAD). Selon [25], qui est l'application de IPv6 dans les réseaux cellulaires 2G et 3G, mentionne que la responsabilité de la distribution des identificateurs uniques est attribuée GGSN. En effet, cette pratique permet d'assurer l'unicité des identificateurs uniques distribués dans le réseau d'où l'unicité absolue des adresses IPv6 créées et l'élimination du DAD. Si la possibilité qu'il y ait deux mêmes adresses IPv6 dans un même réseau devient grande, nous pourrions imposer le DAD avant de pénétrer dans le nouveau réseau. Ainsi, l'adresse sera créée et envoyée au prochain routeur d'accès pour vérifier son unicité. Le routeur d'accès répondra à l'unité

mobile dans son réseau actuel. Dans le cas où l'unité mobile ne suit plus son profil de mobilité, une création d'adresse va être faite selon le protocole de gestion de mobilité par défaut.

3.3.5 Détails de la relève

Maintenant que nous avons détaillé tous les éléments du modèle de gestion de mobilité, nous parlerons de la manière dont le modèle les utilisera pour effectuer une relève. Une relève sera enclenchée après la réception d'informations de la couche 2 « *L2 Trigger* ». Cette information de la couche 2 contient l'identifiant de ce nouveau point d'accès (adresse MAC, EUI-64 etc.). Avec l'identifiant nous pouvons vérifier dans le profil de mobilité si le point d'accès fait partie du profil. Si ce point d'accès fait partie du profil, nous utilisons le préfixe de sous-réseau qui lui est associé (voir Tableau 3.2) sans attendre le *Routing Advertisement* du routeur d'accès pour créer une COA dans le prochain sous-réseau (*NCOA*). A ce moment là, le MN va commencer par envoyer un BU à ses CN et HA avec son adresse dans le prochain réseau (*NCOA*). En même temps, un *Fast Binding Update* (FBU) sera envoyé au routeur d'accès précédent (PAR) pour redirectionner les paquets destinés au MN vers son nouveau routeur d'accès (NAR). Le FBU contient la prochaine adresse du MN; le PAR l'utilisera donc pour redirectionner les paquets. Le PAR répondra à cette demande d'expédition de paquets par un *Fast Binding Update Acknowledgement* (FBACK) comme accusé de réception. A ce moment là, le MN va effectuer une relève à la couche 2 et commencera à communiquer sur le nouveau réseau avec sa nouvelle adresse. La Figure 3.7 illustre les détails de la relève SMIPv6.

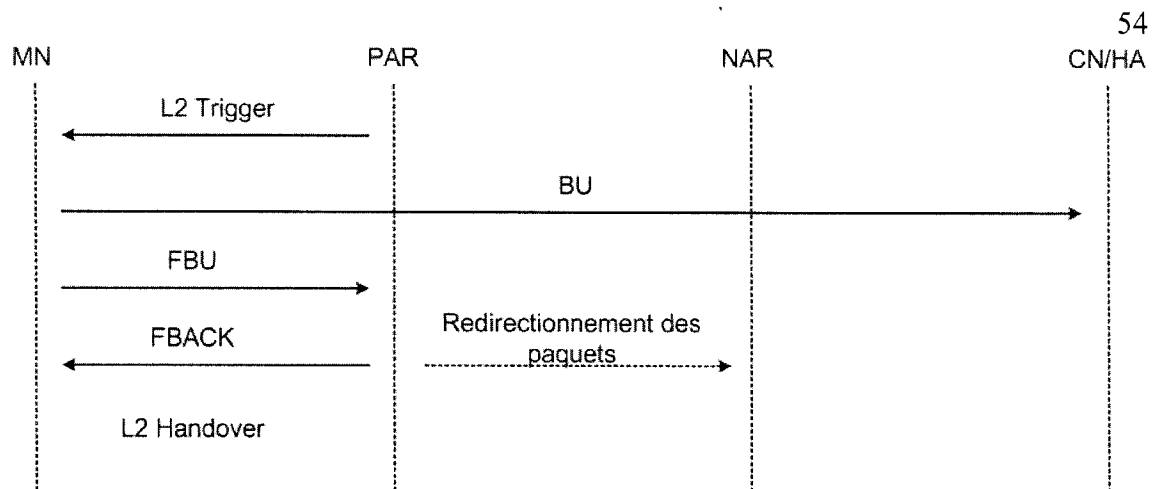


Figure 3.7 Détails de la relève SMIPv6

Dans le cas où le FBACK n'est pas reçu après un certain temps, une relève normale à la couche 2 entre les deux points d'accès est effectuée. Après réception de l'information sur le nouveau préfixe de sous-réseau, une nouvelle COA est créée et des BUs avec cette adresse sont envoyés au HA, PAR et CN. La Figure 3.8 illustre une relève SMIPv6 qui a échoué.

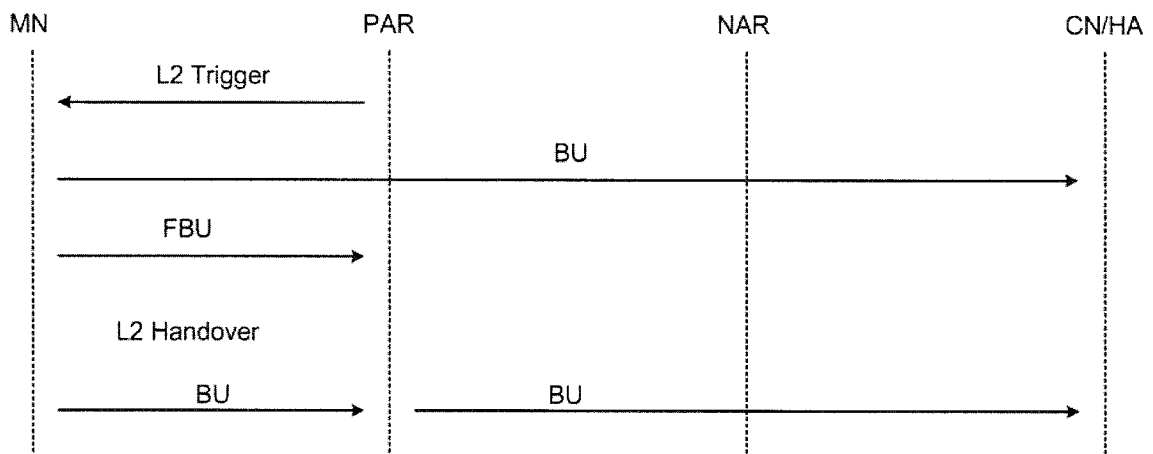


Figure 3.8 Détails de la relève SMIPv6 échouée

Si ce point d'accès ne fait pas partie du profil de mobilité, le MN ne peut pas avoir le préfixe de sous-réseau pour créer sa nouvelle COA. Il effectue donc une relève du protocole de gestion de mobilité par défaut, étant MIPv6 dans notre cas. En effet, il attend un *Routing Advertisement* pour créer sa nouvelle adresse, effectue une relève à la couche 2 et envoie des BU à ses CN et HA avec sa nouvelle COA. La Figure 3.9 illustre les détails de la relève MIPv6.

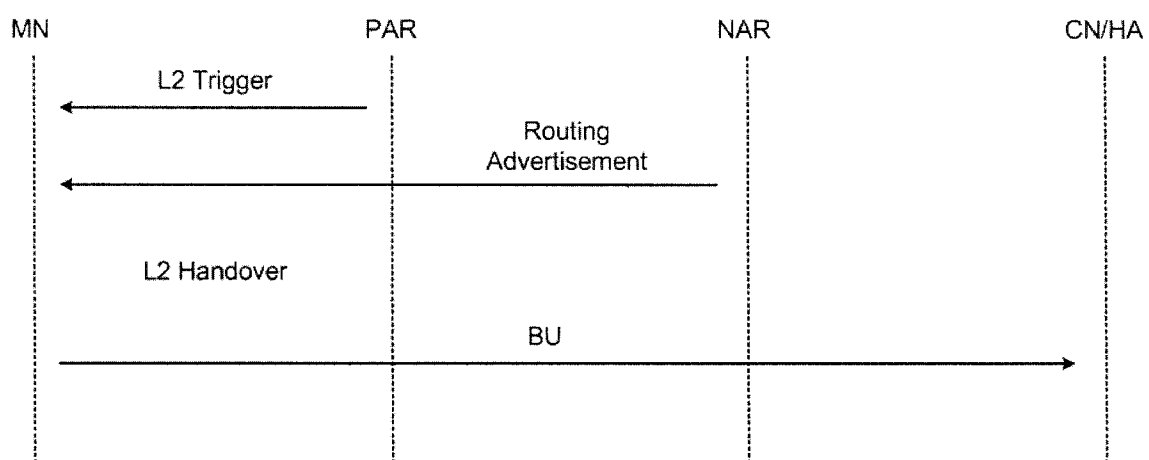


Figure 3.9 Détails de la relève MIPv6

3.4 Description des algorithmes

Une fois l'entité de gestion de mobilité définie, nous allons détailler les algorithmes qui régissent son comportement. En effet, chaque unité mobile répond à différents événements qui se produisent dans l'environnement sont :

1. communication avec un nouveau CN
2. réception du signal d'un nouveau point d'accès
3. expiration du temps de validité de la liste des stations de base
4. expiration du temps de validité de la dernière mise à jour envoyée.

Notre unité mobile réagit différemment à ces événements selon l'état où elle se trouve. La Figure 3.10 illustre les différents états de l'unité mobile. L'unité mobile suit son profil de mobilité la plupart du temps. À partir de cet état, elle peut soit continuer à

suivre soit ne plus suivre son profil. Une fois que l'unité mobile aura quitté son profil, elle pourra toujours y retourner. Nous ne considérons pas le cas où l'unité quitte la zone de couverture car à ce moment l'unité mobile ne peut plus communiquer. Une fonction sera utilisée pour vérifier l'état du MN; les réponses possibles sont dehors (Hors du Profil) et dedans (Suit le Profil).

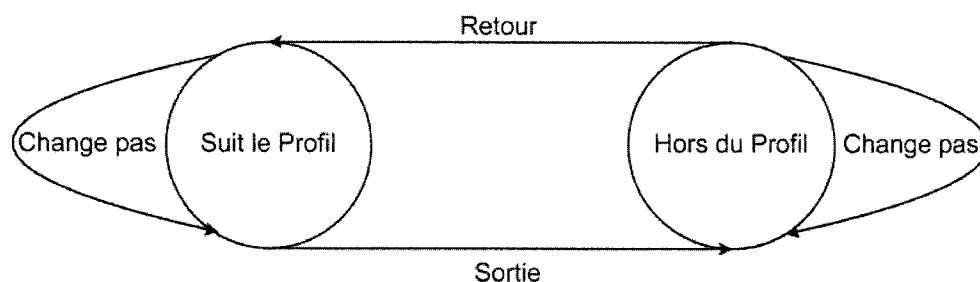


Figure 3.10 Diagramme d'état de l'unité mobile

L'état de l'unité mobile affecte principalement l'envoi des mises à jour des associations (BU). En effet, une unité mobile qui suit son profil va envoyer un BU avec sa future adresse (NCOA) provenant de son profil de mobilité et une durée de vie correspondant à cette adresse qu'on appellera *profile_end_time + delay*.

Or si l'unité mobile ne suit plus son profil de mobilité, à ce moment l'unité mobile utilisera la méthode de mise à jour des associations par défaut en envoyant un BU avec son adresse actuelle à intervalle défini qu'on appellera *max_rate*. Nous allons décrire la réponse de l'unité mobile à chaque événement en considérant son état actuel.

Communication avec un nouveau CN

Lors d'une communication avec un nouveau CN, l'unité mobile rajoutera ce CN à sa liste de correspondants et vérifie son état actuel. Si le MN suit son profil, il envoie un BU avec son adresse actuelle ainsi que la durée de vie de cette adresse *profile_end_time + delay*. Si le MN a quitté son profil, il envoie un BU avec son adresse

actuelle et une durée de vie *max_rate*. Après l'envoi du BU, la liste des BU est mise à jour.

Expiration de la durée de vie d'une association (BU Timeout)

Une unité mobile maintient une liste de BU envoyés avec leur durée de vie. Lorsqu'une association envoyée expire (*BU Timeout*), l'unité mobile doit envoyer de nouveaux BU à ses correspondants et mettre à jour la liste de BU. En effet, le MN enverra un BU à tous ses CN avec son adresse actuelle et une durée de vie *max_rate*.

Réception du signal d'un nouveau point d'accès

Lorsque le MN reçoit un signal diffusé par une station de base, il se trouve dans deux situations possibles. La première : le MN ère dans la zone de couverture du même réseau et met à jour la liste de stations de base (BS). La deuxième : il a pénétré dans la zone de couverture d'un nouveau réseau. À ce moment là, il vérifie si cette BS fait partie de son profil de mobilité. Si le nouveau réseau fait partie de son profil de mobilité, il commencera une relève SMIPv6. Il utilisera le préfixe de sous-réseau qui est associé à cette station de base dans le profil de mobilité (voir Tableau 3.2) pour créer une COA dans le prochain sous-réseau (*NCOA*). À ce moment là, le MN va commencer par envoyer un BU à ses CN et HA avec son adresse dans le prochain réseau (*NCOA*). En même temps, un *Fast Binding Update* (FBU) sera envoyé au routeur d'accès précédent (PAR) pour redirectionner les paquets destinés au MN vers son nouveau routeur d'accès (NAR). Le PAR répondra à cette demande d'expédition de paquets par un *Fast Binding Update Acknowledgement* (FBACK) pour confirmer. À ce moment là, le MN va effectuer une relève à la couche 2 et commencera à communiquer sur le nouveau réseau avec sa nouvelle adresse. Si le nouveau réseau ne fait pas partie du profil, il effectuera une relève normale (voir section 3.3.5). En effet, le MN attendra de recevoir un RA pour avoir le préfixe de sous-réseau, créer une nouvelle COA, effectuer un L2 Handover et envoyer les BU au HA et CN.

Expiration de la durée de vie d'une station de base (BS Timeout)

Une unité mobile maintient une liste de BS visitées. Lorsqu'une BS n'a pas envoyé de RA pendant un certain temps, cette entrée dans la liste de BS sera supprimée. Si cette entrée concerne une ancienne BS, elle est supprimée tout simplement. Si cette entrée concerne la station de base actuelle, à ce moment là on considère que le contact est perdu avec cette BS. S'il n'existe aucune autre BS dans la liste, nous considérons que l'unité mobile a quitté la zone de couverture et la MN reste en attente d'un nouveau signal. Si par contre il y a une autre BS dans la liste, nous vérifions si cette BS fait partie du profil de mobilité du MN. Si le nouveau réseau fait partie de son profil de mobilité, il commencera une relève SMIPv6. Si le nouveau réseau ne fait pas partie du profil, il effectuera une relève normale (voir section 3.3.5).

La Figure 3.11 et la Figure 3.12 présentent les diagrammes d'états des algorithmes détaillés auparavant.

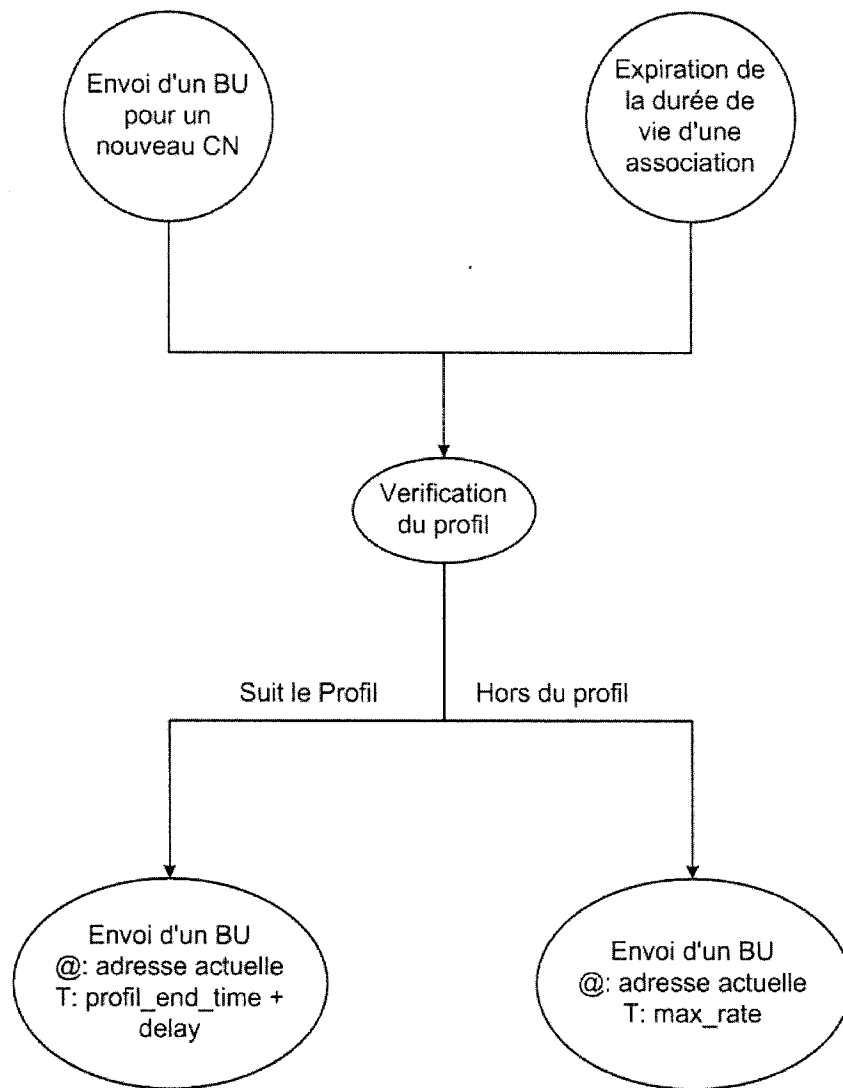


Figure 3.11 Envoi d'un BU pour un nouveau CN et expiration d'un BU

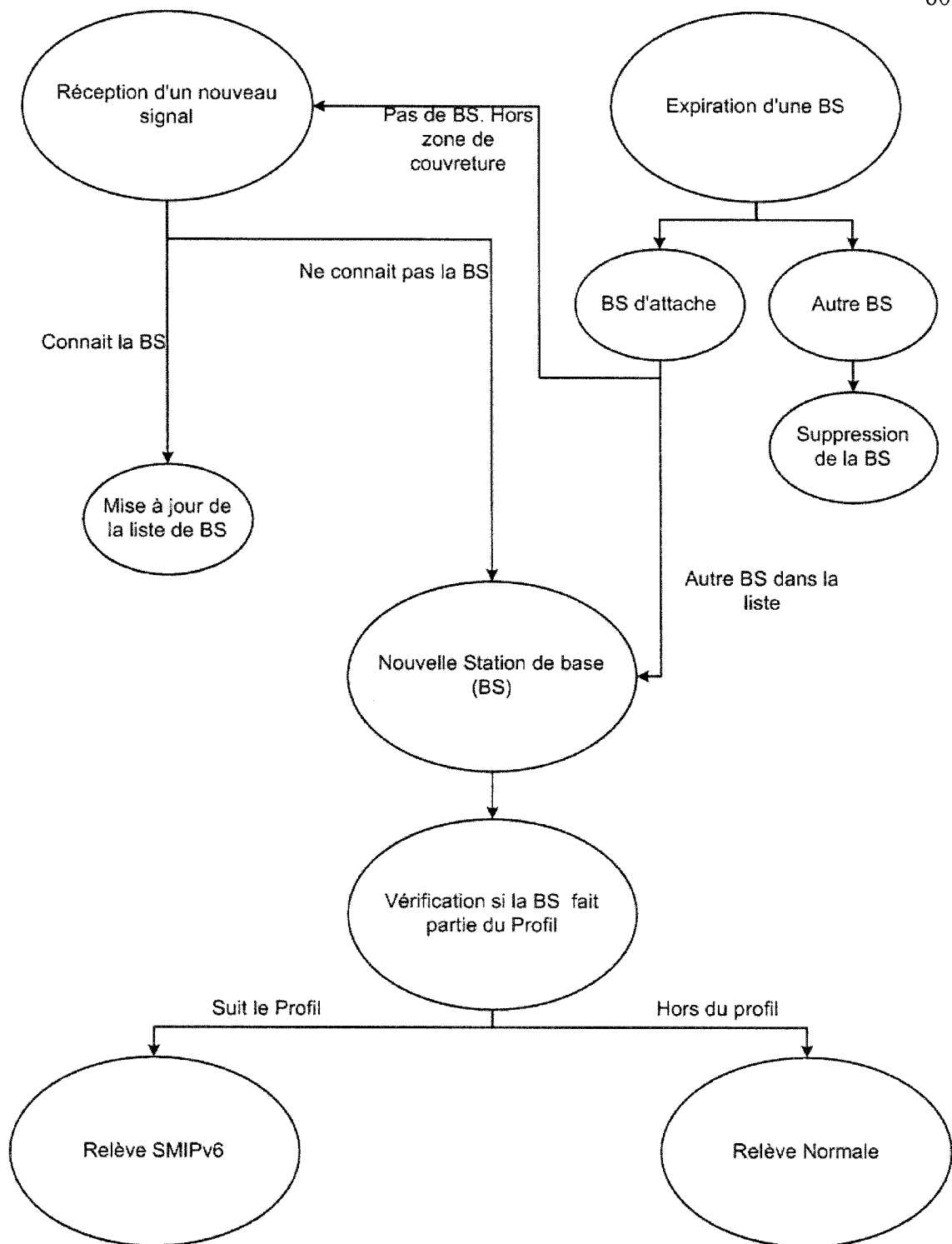


Figure 3.12 Réception d'un RA, expiration du profil ou expiration d'une BS

3.5 Analyse mathématique

Après avoir détaillé tous les composants et mécanismes, nous allons faire une étude analytique de la performance du protocole SMIPv6 comparé aux protocoles MIPv6 et FMIPv6. En effet, cette étude analytique permet de montrer les avantages de SMIPv6 en terme de réduction du délai de la relève et de perte de paquets. Pour estimer les délais, nous considérons l'équation (3.1) :

$$D = D1 + D2 \quad (3.1)$$

où D1 est le délai de propagation sur le lien et D2 le délai de traitement des nœuds.

Soit les délais suivants :

T1 (a, b) : délai d'un lien filaire (a, b)

T2 (n) : délai de traitement à un nœud n

T3 (a, b) : délai d'un lien radio (a, b)

Nous considérons la topologie de la Figure 3.13 composée d'un nœud mobile MN, de deux routeurs d'accès PAR et NAR, de quatre routeurs du réseau cœur N1, N2, N3, N4, d'un correspondant CN et d'un Home Agent HA.

Tous les délais des liens filaires ont la valeur f sauf le lien T1 (N1, N2) qui a la valeur F . Ce lien sert à représenter la mobilité globale et la mobilité locale. En effet, si ce lien a une valeur élevée (au dessus de 100 ms), nous considérons une mobilité globale et si le lien a une valeur faible (moins de 10 ms), nous considérons une mobilité locale du MN. Le délai d'un lien radio aura la valeur d , sachant que le délai de transmission d'un lien radio varie avec la distance entre le MN et son point d'accès. Cependant, l'échange de paquet lors d'une relève s'effectue toujours dans la zone de chevauchement (à l'extrémité de la cellule). Le délai de la relève à la couche 2 a une valeur h .

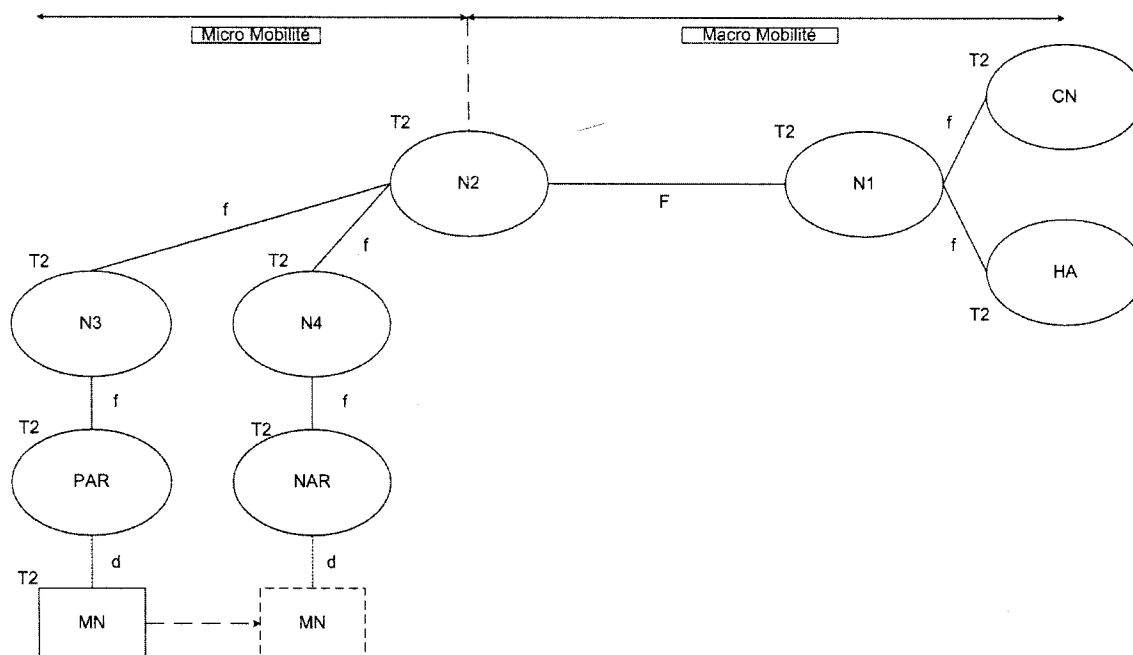


Figure 3.13 Topologie utilisée pour l'étude analytique

Nous voulons comparer le délai de la relève entre les différents protocoles et la perte de paquets. Le délai de la relève représente le délai entre le moment où le dernier paquet de données est reçu à travers l'ancien point d'accès et le moment où le premier paquet de données est reçu à travers le nouveau point d'accès. Le nombre de paquets perdus représente le nombre de paquets que le MN est dans l'impossibilité de recevoir puisqu'il a changé de point d'accès, paquets qui n'ont pas pu être récupérés. Le délai de la signalisation durant une relève est le temps durant lequel le MN est en train d'établir une relève. Ce délai commence par le premier paquet de signalisation et se termine au moment où la table de routage du CN est mise à jour.

- **Cas d'une relève MIPv6**

La Figure 3.14 donne le détail d'une relève avec le protocole MIPv6. Le Tableau 3.3 détaille les échanges dans l'ordre chronologique avec le délai engendré par chaque événement.

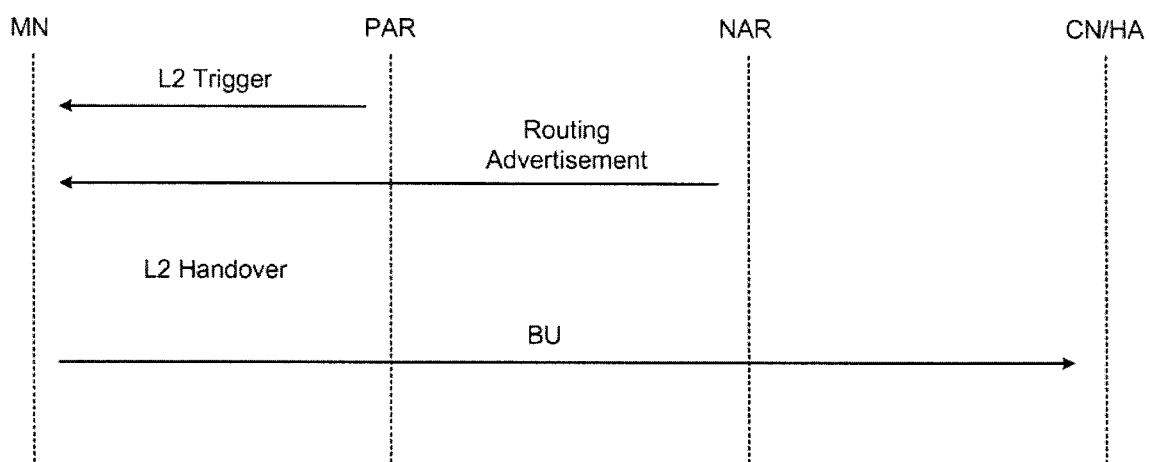


Figure 3.14 Détails d'une relève MIPv6

Tableau 3.3 Détails chronologiques d'une relève MIPv6

Temps	Événement	Délai	
$t = 0$	Réception du RA	T2	(1)
$t = T2$	L2 Handover	h	(2)
$t = T2 + h$	Envoi d'un BU aux HA/CN	$3f + F + d + 6T2$	(3)
$t = 3f + F + d + h + 7T2$	Paquets envoyés par CN à la NCOA	$3f + F + d + 6T2$	(4)
$t = 6f + 2F + 2d + h + 13T2$	Réception des paquets par le MN		(5)

Dernier paquet reçu à travers NAR à $t = T2$ (d'après 2)

Premier paquet reçu à travers PAR à $t = 6f + 2F + 2d + h + 13T_2$ (d'après 5)

Le délai de la relève est donc : $6f + 2F + 2d + h + 12T_2$ (5) – (2)

À partir du moment où le MN effectue la relève et le moment où le CN envoie les paquets à la NCOA, les paquets sont envoyés à son ancienne COA et perdus par conséquent. Le nombre de paquets perdus est : $(3f + F + d + h + 5T_2) \times \text{débit}$.

Le délai de la signalisation commence au moment où le RA est reçu par le MN (2) et le moment où le BU est reçu par le CN (3). Ainsi, le délai de signalisation est : $3f + F + d + h + 5T_2$.

• Cas d'une relève FMIPv6

La Figure 3.15 donne le détail d'une relève avec le protocole FMIPv6. Le Tableau 3.4 détaille les échanges dans l'ordre chronologique, avec le délai engendré par chaque événement.

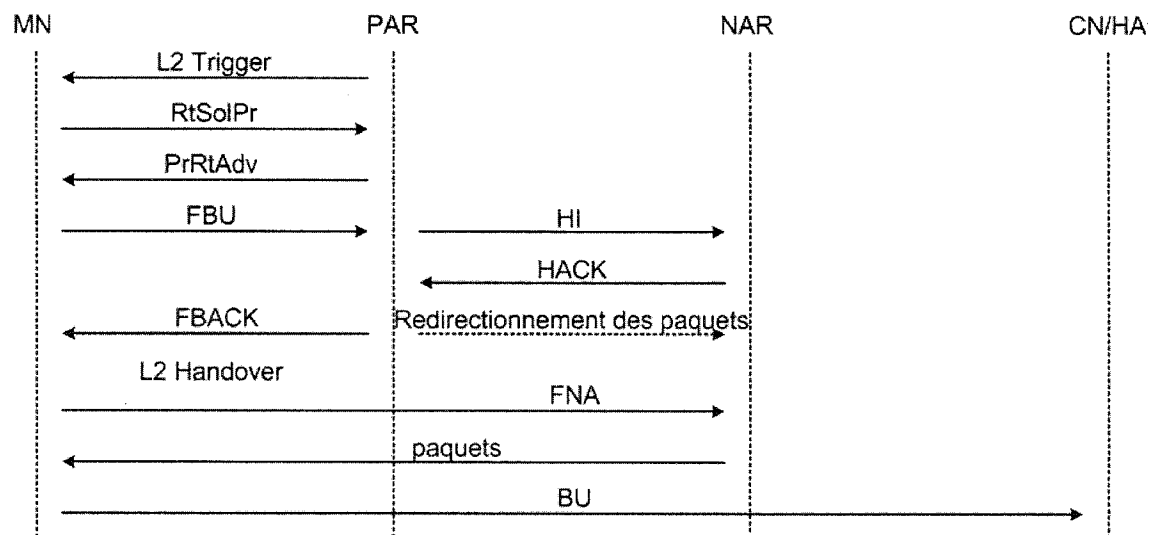


Figure 3.15 Détails d'une relève FMIPv6

Dernier paquet reçu à travers NAR à $t = 4d + 8f + 18T_2$ (d'après 9)

Premier paquet reçu à travers PAR à $t = \max(6d + 8f + h + 22T_2, 12f + 4d + 23T_2)$ (d'après 12)

Le délai de la relève est donc : $\max(2d + h + 4T_2, 4f + 5T_2)$

Il n'y a pas de paquets perdus vu que le PAR commence à redirectionner les paquets vers le NAR avant d'effectuer la relève et les paquets sont gardés dans une mémoire tampon en attendant le L2 Handover. Les paquets sont ensuite envoyés au MN après la réception du FNA. La perte de paquet est nulle.

Le délai de signalisation est important vu que le L2 Trigger est reçu par le MN à T_2 (2) et que le BU est reçu par les CN et HA à $11f + 5d + F + h + 24T_2$ (13). Ainsi, le délai de signalisation est de $11F + 5d + F + h + 23 T_2$.

Tableau 3.4 Détails chronologiques d'une relève FMIPv6

Temps	Événement	Délai	
$t = 0$	L2 Trigger	T_2	(1)
$t = T_2$	RtSolPr	$d + 2T_2$	(2)
$t = d + 2T_2$	PrRtAdv	$d + 2T_2$	(3)
$t = 2d + 4T_2$	FBU	$d + 2T_2$	(4)
$t = 3d + 6T_2$	HI	$4f + 5T_2$	(5)
$t = 3d + 4f + 11T_2$	HACK	$4f + 5T_2$	(6)
$t = 3d + 8f + 16T_2$	FBACK	$d + 2T_2$	(7)
$t = 3d + 8f + 16T_2$	Redirectionnement des paquets au PAR	$4f + 5T_2$	(8)
$t = 4d + 8f + 18T_2$	L2 Handover	h	(9)
$t = 4d + 8f + h + 18T_2$	FNA	$d + 2T_2$	(10)
$t = 4d + 8f + h + 18T_2$	Envoi d'un BU aux HA/CN	$3f + F + d + 6T_2$	
$t = \max(5d + 8f + h + 20T_2, 12f + 3d + 21T_2)$	PAR envoie les paquets au MN	$d + 2T_2$	(11)
$t = \max(6d + 8f + h + 22T_2, 12f + 4d + 23T_2)$	Réception des paquets par le MN		(12)

$t = 11f + 5d + F + h + 24T2$	Réception des BU		(13)
-------------------------------	------------------	--	------

- **Cas d'une relève SMIPv6**

La Figure 3.16 donne les détails d'une relève avec le protocole SMIPv6. Le Tableau 3.5 détaille les échanges dans l'ordre chronologique avec le délai engendré par chaque événement.

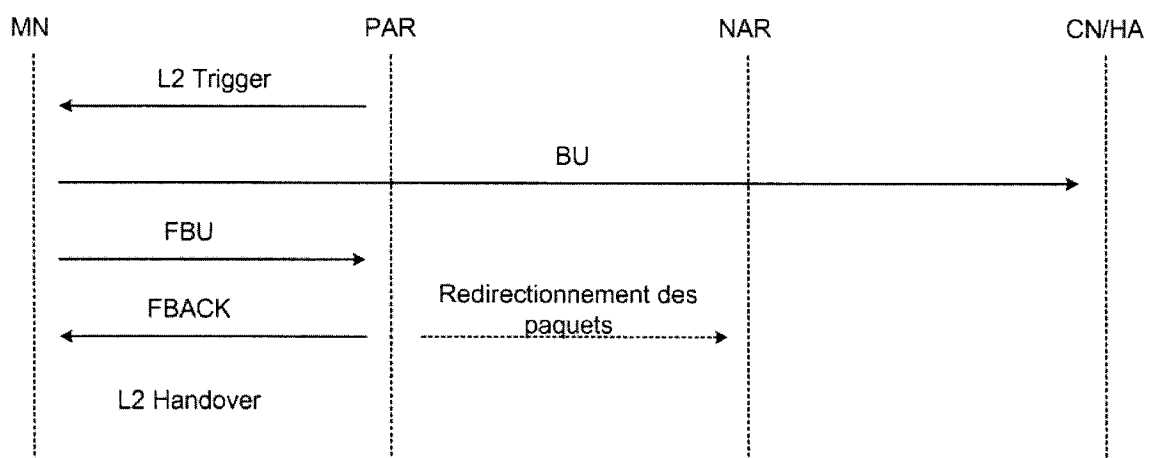


Figure 3.16 Détails d'une relève SMIPv6

Tableau 3.5 Détails chronologiques d'une relève SMIPv6

Temps	Événement	Délai	
$t = 0$	L2 Trigger	$T2$	(1)
$t = T2$	FBU	$d + 2T2$	(2)
$t = T2$	BU	$3f + F + d + 6T2$	(3)
$t = d + 3T2$	FBACK	$d + 2T2$	(4)
$t = d + 3T2$	Redirectionnement des paquets	$4f + d + 6T2$	(5)
$t = 2d + 5T2$	L2 Handover	h	(6)
$t = d + 3f + L + 6T2$	CN envoie les paquets à NCOA	$3f + F + d + 6T2$	(7)

$t = 4f + 2d + 9T2$	Paquets redirectionnés reçus		(8)
$t = 6f + 2F + 2d + 12T2$	Paquets du CN reçus		(9)

Dernier paquet reçu à travers NAR à $t = 2d + 5T2$ (d'après 6)

Premier paquet reçu à travers PAR à $t = \min(2d + 4f + 9T2, 2d + 6f + 2F + 12T2)$

Le délai de la relève est donc : $4f + 4T2$

Il n'y a pas de paquets perdus vu que le PAR commence à redirectionner les paquets vers le NAR avant d'effectuer la relève. Le MN se trouve dans son nouveau réseau avant même que les paquets redirectionnés ou envoyés par le CN n'y arrive. En effet, les paquets redirectionnés arrivent les premiers à $4f + 2d + 6T2$ et le MN se trouve dans son nouveau réseau à $2d + h + 5T2$. Ainsi, le MN se trouve à son nouveau réseau $4f - h + T2$ avant les paquets. $4f - h + T2$ est une valeur positive vu que h (le délai de la relève à la couche 2) avoisine 0 et que $T2$ est relativement petit. Ainsi, la perte de paquet est nulle.

Le délai de signalisation correspond au moment où le MN reçoit le L2 Trigger (2) et le moment où le BU est reçu par le CN/HA (7). Ainsi, le délai de signalisation est : $3f + F + d + 5T2$.

- **Comparaison des résultats**

Le Tableau 3.6 récapitule les résultats trouvés auparavant pour les protocoles MIPv6, FMIPv6 et SMIPv6.

Tableau 3.6 Comparaison de performance des protocoles

MIPv6, FMIPv6 et SMIPv6

	Délai de la relève	Perte de paquets	Délai de signalisation
MIPv6	$6f + 2F + 2d + h + 12T2$	$3f + F + d + h + 5T2$	$3f + F + d + h + 5T2$
FMIPv6	$\max(2d + h + 4T2, 4f + 5T2)$	0	$11f + 5d + F + h + 24T2$

SMIPv6	$4f + 4T2$	0	$3f + F + d + 5T2$
---------------	------------	---	--------------------

En comparant le délai de la relève, on remarque que MIPv6 est le protocole qui a le plus grand délai. FMIPv6 a un délai de relève supérieur comparé à celui de SMIPv6 dans tous les cas. En effet, SMIPv6 offre une relève avec le plus petit délai de relève parmi les protocoles comparés.

En ce qui concerne le nombre de paquets perdus, MIPv6 est celui qui subit le plus grand nombre de pertes de tous. FMIPv6 et SMIPv6 ne subissent pas de perte de paquets. Cependant, FMIPv6 doit utiliser une mémoire tampon pour sauvegarder les paquets redirectionnés pendant la relève, ce qui n'est pas une solution très évolutive. En effet, dans le cas où une cellule est pénétrée par plusieurs unités mobiles en même temps, il serait difficile de sauvegarder tous leurs paquets durant la relève.

Finalement, le délai de signalisation nous donne une idée sur le temps nécessaire pour établir une relève. En effet, tous ces messages de signalisation doivent être échangés quand le MN se trouve encore dans la zone de chevauchement. Ainsi, plus cette zone est petite et la vitesse de déplacement du MN est grande, plus le temps pour établir cette signalisation est petit. Nous pouvons donc conclure que, même si FMIPv6 peut établir une relève sans perte de paquets avec un délai de relève plus ou moins comparable à celui de SMIPv6, il a un temps de signalisation très important qui pourrait engendrer l'échec de cette relève dans certaines circonstances.

CHAPITRE IV

IMPLÉMENTATION ET RÉSULTATS

Après avoir conçu notre protocole de gestion de mobilité, nous procédons dans ce chapitre à son implémentation. Comme le cycle de développement complet d'un protocole prenant en compte tous les aspects de spécifications peut s'avérer fort long et nous éloigner de nos principaux objectifs, nous préférons nous focaliser sur une partie définie du travail qui permet de valider l'ensemble du travail présenté. Nous avons choisi *Network Simulator* (NS), présenté en annexe, pour réaliser les simulations de réseau IPv6. Puis, nous décrirons l'implémentation du protocole MIPv6 initial ainsi que les améliorations encodées pour aboutir au protocole *Seamless Mobile IPv6*. Puis, nous suivrons le déroulement de notre implémentation dans une configuration donnée. Pour finir, et après avoir défini notre plan d'expérience, nous présenterons les résultats de simulations.

4.1 Implémentation du protocole proposé : Seamless Mobile IPv6

Le protocole proposé, Seamless Mobile IPv6 (SMIPv6), est principalement basé sur le module MobiWan (voir annexe). Nous avons ajouté la notion de profil de mobilité, modifié la structure des paquets MIPv6, ajouté de nouveaux paquets pour permettre une mobilité sans coupure et implémenté notre protocole.

4.1.1 Correction apportée à MobiWan

Dans l'implémentation originale de MobiWan, pour des raisons d'optimisation du temps de simulation les stations de bases n'envoient pas de *Router Advertisements* par défaut. Cette configuration permet de réduire le temps de simulation pour des grandes topologies où les stations de base n'ont pas de MN avec qui communiquer.

Ainsi, les stations de base commencent à diffuser des RA seulement quand un MN envoie une demande de RA. Cependant, cette configuration peut causer des problèmes. En effet, lorsque les BS n'envoient pas de RA par défaut, le MN doit absolument envoyer une demande de RA (Solicitation) pour détecter une nouvelle BS lorsqu'il ne reçoit plus de RA. À ce moment là, le MN a perdu la connexion avec son ancienne BS. Cette méthode élimine toute possibilité de relèvement sans coupure puisque le MN ne peut pas découvrir de nouvelles BS pendant qu'il est sous la zone de couverture de sa BS par défaut. Pour résoudre ce problème, nous avons activé la diffusion des *Router Advertisements* par défaut dans toutes les stations de base au début de la simulation.

4.1.2 Profil de mobilité

Pour pouvoir implémenter la notion de Profil de Mobilité, nous fournissons un fichier contenant un profil de mobilité au nœud mobile. Ce dernier le chargera dans sa mémoire dans une liste appelée *my_profile_* composée d'entrées de type *profile_entry* chacune contenant l'identifiant de la cellule *cell_id_*, le préfixe de sous-réseau *subnet_prefix_*, le temps d'entrée *start_time_* et le temps de sortie de cette cellule *end_time_*.

Pour qu'un MN puisse savoir s'il suit son profil ou non, une fonction *status_change* est utilisée pour vérifier si la cellule dans laquelle le MN se trouve fait bien partie du profil de mobilité dans un intervalle $[t - \text{delay}, t + \text{delay}]$, qui est un intervalle incluant le facteur *delay* qui permet d'absorber les retards. Cette fonction, après avoir fait la vérification, inscrira la valeur 1 dans la variable *profile_ok_* et pointera *current_entry* à l'entrée du profil actuel si le MN suit son profil ou inscrira la valeur 0 s'il a quitté son profil. L'entrée de type *profile_entry_* pointée par *current_entry* sert à avoir l'information sur la position de l'unité mobile selon le profil de mobilité.

4.1.3 Nouveaux paquets

Dans notre nouveau protocole, nous avons utilisé la même structure que l'entête d'un paquet Mobile IPv6 traditionnel. Cependant, nous avons introduit deux genres de paquets : FBU et FBACK. Selon notre protocole, un MN qui s'apprête à effectuer une relève sans coupure enverra un *Fast Binding Update* (FBU) qui lui permettra de demander à sa BS actuelle de créer une association entre la COA actuelle et la future COA. Ainsi, tout le trafic sera redirigé vers la prochaine BS. La BS répondra avec un FBACK pour confirmer le début du redirectionnement.

4.1.4 Modifications apportées au protocole MIPv6

Les principales modifications apportées au module MobiWan pour aboutir au protocole proposé sont :

1. envoi d'un BU pour un nouveau CN
2. réception du signal d'un nouveau point d'accès
3. expiration du temps de validité d'une BS
4. expiration du temps de validité d'un BU.

Envoi d'un BU pour un nouveau CN

Lorsqu'un nouveau CN veut communiquer avec un MN, il ne connaît pas toujours sa COA actuelle. Ainsi, il lui envoie les paquets à sa *Home Address*. Le *Home Agent* intercepte le paquet, l'encapsule et l'envoie à la COA actuelle du MN. Le MN reçoit ce paquet par son *Network Agent* (*NetworkMN::recv*). Selon l'entête du paquet, le MN comprend que le paquet est encapsulé et retransmis par un autre routeur d'accès et le décapsule. Le MN vérifie si la source du paquet décapsulé existe dans la liste *Binding Update List*. Si le nœud n'existe pas dans la liste de nœuds auxquels le MN envoie des BU régulièrement, il l'ajoute à cette liste et un BU est envoyé avec la fonction *send_bu_to_new_cn*. Si le CN existe dans la liste, le MN ne fait rien.

Pendant une relève avant que le CN ne reçoive le BU, les paquets seront envoyés à l'ancienne COA du MN et retransmis par l'ancien AR. Nous ne voulons pas avoir à

envoyer autant de BU que de paquets retransmis. En effet, nous vérifions l'existence du CN dans la liste pour savoir si un BU a déjà été envoyé.

La fonction *send_bu_to_new_cn* vérifie si le MN suit son profil avec *status_change* pour voir comment faire son BU. Si le MN suit son profil, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au temps de sortie de la cellule actuelle avec le facteur retard (*current_entry.end_time + delay*), et la minuterie du BU est remontée à *current_entry.end_time + delay*. Dans le cas contraire, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au *max_rate*, et la minuterie du BU est remontée à *max_rate*.

Réception du signal d'un nouveau point d'accès

Dans *Network Simulator*, il n'y a pas la notion d'informations provenant de la couche 2. Ainsi, nous allons simuler cet aspect en utilisant les *Router Advertisements* des routeurs d'accès à la couche 3 comme étant des informations de la couche 2. En effet, ces informations de la couche 2 provenant des points d'accès permettent la détection de mouvement du MN mais les RA de la couche 3 ne sont pas aussi fréquents que les RA de la couche 2. Pour cela, nous allons modifier l'intervalle d'envoi des RA de la couche 3.

Ces *Router Advertisements* sont reçus par la fonction *recv_ads* du MN. Une fois un RA reçu, le MN vérifie si nous connaissons cette BS. Si le MN connaît cette BS, l'entrée de cette BS est simplement mise à jour dans la *Base Station List*. Dans le cas contraire, le MN déclare que c'est une nouvelle BS, crée une nouvelle entrée pour cette BS, déclare sa COA perdue (*coalost = TRUE*) et appelle la fonction *reg* pour enregistrer le MN avec cette BS. La fonction *reg* commence par vérifier si *coalost* est activé. Si c'est le cas, la fonction continue de vérifier si le MN était attaché à une autre BS pour savoir si une relève SMIP est possible. Dans le cas contraire, une relève normale est faite et la COA actuelle est changée.

Dans le cas d'une relève normale, le MN vérifie si cette BS fait partie de son profil. Si elle fait partie de son profil, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au temps de sortie de la cellule actuelle avec le facteur retard ($current_entry.end_time_ + delay$), et la minuterie du BU est remontée à $current_entry.end_time_ + delay$. Dans le cas contraire, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au max_rate , et la minuterie du BU est remontée à max_rate .

Dans le cas d'une relève SMIP, un FBU est envoyé à la BS actuelle avec la COA actuelle et la future COA. Le MN envoie aussi des BU au HA et CN avec la prochaine COA du MN provenant du profil de mobilité et une durée de vie correspondant à la durée de vie de cette COA. En recevant le FBU, la BS actuelle ajoute une association à sa table de routage entre la COA actuelle et la future COA du MN pour permettre de rediriger le trafic. La BS répond avec un FBACK à l'adresse actuelle du MN pour confirmer le redirectionnement des paquets. Lorsque le MN reçoit le FBACK par la fonction *recv_back*, il effectue une relève directement. Une nouvelle minuterie *SMIP_Timer* a été implémentée dans le cas où le FBACK n'est pas reçu après 10 msec, une relève normale est déclenchée.

Expiration du temps de validité d'une BS

Lorsque la BS actuelle n'envoie plus de RA pendant un certain temps, la minuterie de cette BS dans le MN expire comme dans MobiWan. Une nouvelle BS est choisie dans la liste et le MN vérifie s'il suit son profil avec *status_change*. Si le MN suit son profil, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au temps de sortie de la cellule ($current_entry.end_time_$) avec le facteur retard ($current_entry.end_time_ + delay$), et la minuterie du BU est remontée à $current_entry.end_time_ + delay$. Dans le cas contraire, un BU est envoyé avec la COA actuelle et une durée de vie correspondant au max_rate , et la minuterie du BU est remontée à max_rate .

Expiration du temps de validité d'un BU

Lorsque la minuterie d'un BU expire, la fonction *MNAgent::reg* est appelée. Normalement, un BU n'expire que dans le cas où le MN ne suit plus son profil. Dans le cas où le MN suit son profil, un nouveau BU est toujours envoyé avant que l'ancien BU n'expire (le dernier BU expire en considérant le facteur retard *delay*). Dans ce cas là, puisque le MN ne suit pas son profil, on envoie un BU avec la COA actuelle et une durée de vie *max_rate*.

4.2 Fonctionnement du modèle de gestion de mobilité

Dans cette section, nous allons passer en détails quelques scénarios qui permettent de démontrer le fonctionnement de notre modèle de gestion de mobilité basé sur les patrons de mobilité des usagers. Nous supposons que nos scénarios ont une topologie identique illustrée à la Figure 4.1.

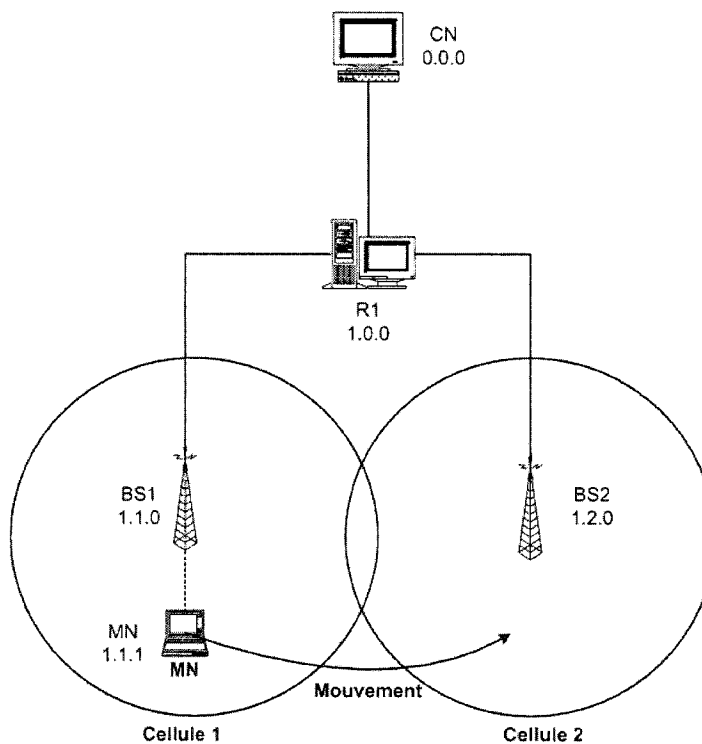


Figure 4.1 Topologie utilisée pour les scénarios

Lors de ce scénario, l'unité mobile se déplace à vitesse constante de 1,38 m/s (5 Km/h). La couverture d'une cellule sera d'un rayon de 250 m. L'unité mobile commence son trajet à la cellule 1 à 0:00 à la position $x = 50$ m. Le mobile parcourra 450 m avant d'arriver à la frontière entre la cellule 1 et 2; ainsi, il devrait effectuer une relève à 0:05:26. Le scénario dure 10 min pour pouvoir visualiser le comportement du MN durant toute la relève. Nous considérons que le retard de notre MN est de 1min; ainsi, la variable *delay* sera égale à 1.

Cette topologie se compose de deux parties : une partie fixe et une partie mobile. Nous utiliserons des adresses IP de la forme x.y.z où x.y est le préfixe du sous-réseau et z l'identifiant unique de l'unité mobile. La partie fixe se constitue d'un nœud correspondant CN (Identifiant 0), d'un routeur R1 et de deux stations de base BS1 (Identifiant 1) et BS2 (Identifiant 2) pour interfacer entre la partie fixe et la partie mobile du réseau. Les stations de base appartiennent à des réseaux différents (préfixes sous-réseau 1.1 et 1.2). La partie mobile se constitue d'une unité mobile MN (Identifiant 4). L'unité mobile se déplace de la cellule 1 à la cellule 2. Cette unité mobile suit un profil de mobilité illustré à la Figure 4.2.

Cellule	Préfixe réseau	te entrée (s)	ts sortie (s)
1	1.1	0:00	0:05
2	1.2	0:05	0:10

Figure 4.2 Profil de mobilité utilisé dans les scénarios

Scénario 1 : Dans ce scénario, l'unité mobile va respecter son profil de mobilité (voir Figure 4.3).

1. À 0 :00, le MN vient de se mettre en marche et il détecte le signal de la cellule 1. Cette BS fait partie du profil de mobilité du MN; ainsi, le MN crée une nouvelle COA pour ce sous-réseau. Il se connecte à cette nouvelle BS et envoie un BU

avec son adresse actuelle (1.1.4) et une durée de vie $current_entry.end_time + delay$ (0:06).

- À 0:05:26, il reçoit le signal de la cellule 2. La cellule 2 fait partie du profil de mobilité du MN, ainsi le MN crée une nouvelle COA pour ce sous-réseau. Il envoie un BU avec sa future adresse (1.2.4) et une durée de vie $current_entry.end_time + delay$ (0:11). Il envoie aussi un FBU au routeur d'accès actuel pour redirectionner les paquets destinés à 1.1.4 vers 1.2.4. Le FBU traite cette requête et répond avec un FBACK. Dès la réception du FBACK, le MN se connecte au nouveau point d'accès.

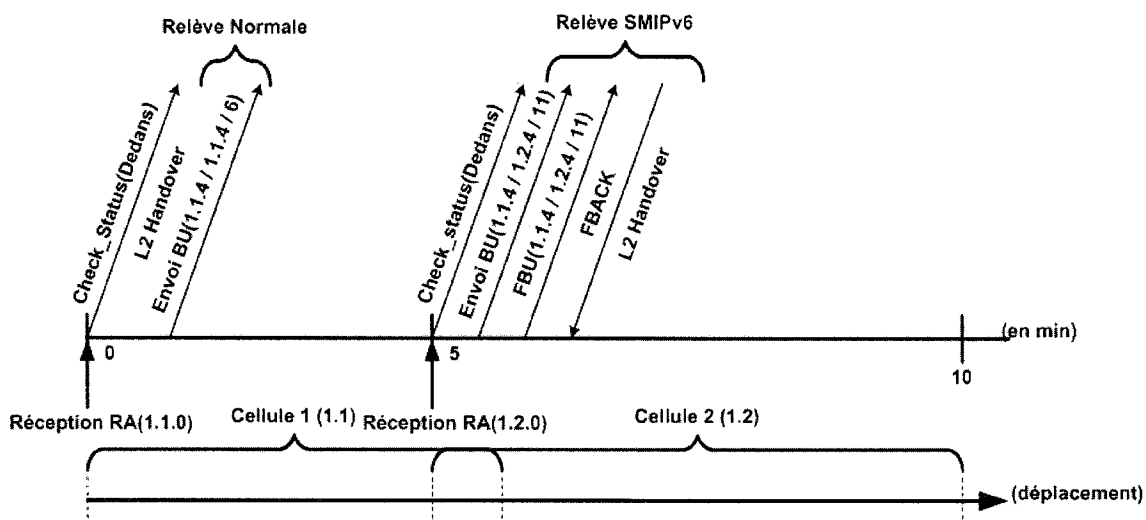


Figure 4.3 Scénario 1

Les Figures 4.4 et 4.5 présentent la variation de la bande passante en fonction du temps pour les protocoles SMIPv6 et MIPv6 respectivement. En comparant les deux figures, nous pouvons remarquer la baisse drastique de la bande passante dans le cas de MIPv6. SMIPv6, par contre, maintient une bande passante constante durant toute la simulation. La variation de la bande passante remarquée dans le cas de SMIPv6 au moment de la relève (0:05), s'explique par le délai de la relève. En effet, la petite baisse de la bande passante s'explique par le fait que les paquets sont redirectionnés et la

hausse de la bande passante s'explique par la réception des paquets redirectionnés et des autres paquets à la nouvelle adresse. MIPv6 subit une baisse de la bande passante dû au fait qu'il n'est muni d'aucun système de redirectionnement des paquets.

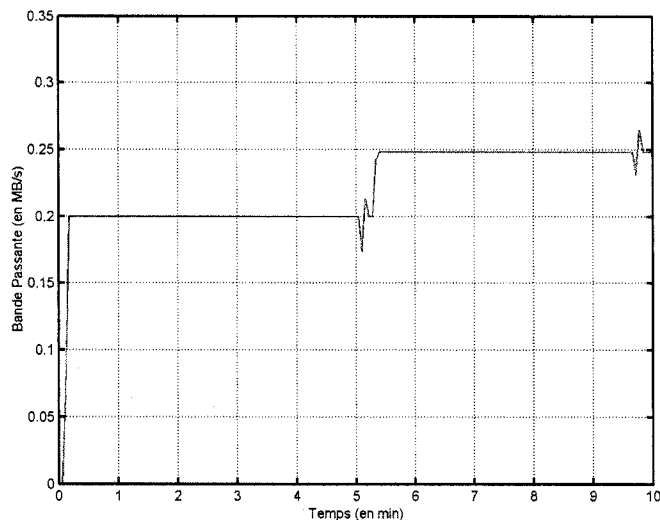


Figure 4.4 Bande passante en fonction du temps – SMIPv6

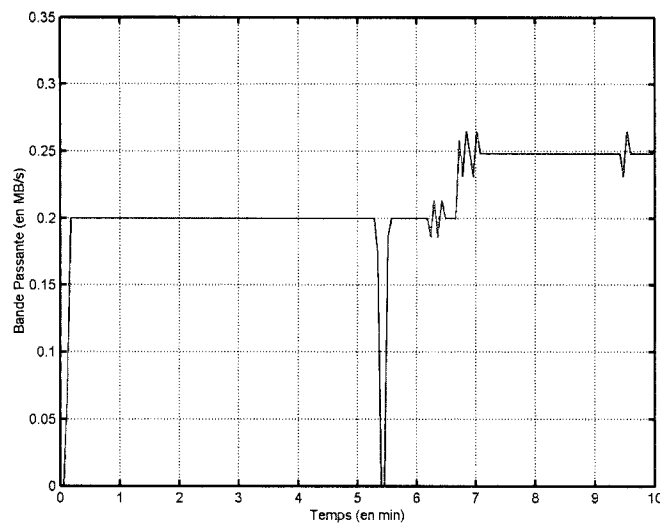


Figure 4.5 Bande passante en fonction du temps – MIPv6

Scénario 2 : Dans ce scénario, l'unité mobile ne respecte pas son profil (voir Figure 4.6). Cette dernière effectuera un retard au moment de pénétrer la cellule 2 (1.2).

1. À 0:00, le MN vient de se mettre en marche et il détecte le signal de la cellule 1. Cette BS fait partie du profil de mobilité du MN; ainsi, le MN crée une nouvelle COA pour ce sous-réseau. Il se connecte à cette nouvelle BS et envoie un BU avec son adresse actuelle (1.1.4) et une durée de vie *current_entry.end_time* + *delay* (0 :06).
2. À 0:06, le BU expirera et va entraîner une vérification de l'état du MN. Le MN ne suit plus son profil et envoie un BU avec son adresse actuelle (1.1.4) et une durée de vie *max_rate* (0:07).
3. À 0:06:30, le MN reçoit le signal de la cellule 2. La cellule 2 fait partie du profil de mobilité du MN, ainsi le MN crée une nouvelle COA pour ce sous-réseau. Il envoie un BU avec sa future adresse (1.2.4) et une durée de vie *current_entry.end_time* + *delay* (0:11). Il envoie aussi un FBU au routeur d'accès actuel pour redirectionner les paquets destinés à 1.1.4 vers 1.2.4. Le FBU traite cette requête et répond avec un FBACK. Dès la réception du FBACK, le MN se connecte au nouveau point d'accès.

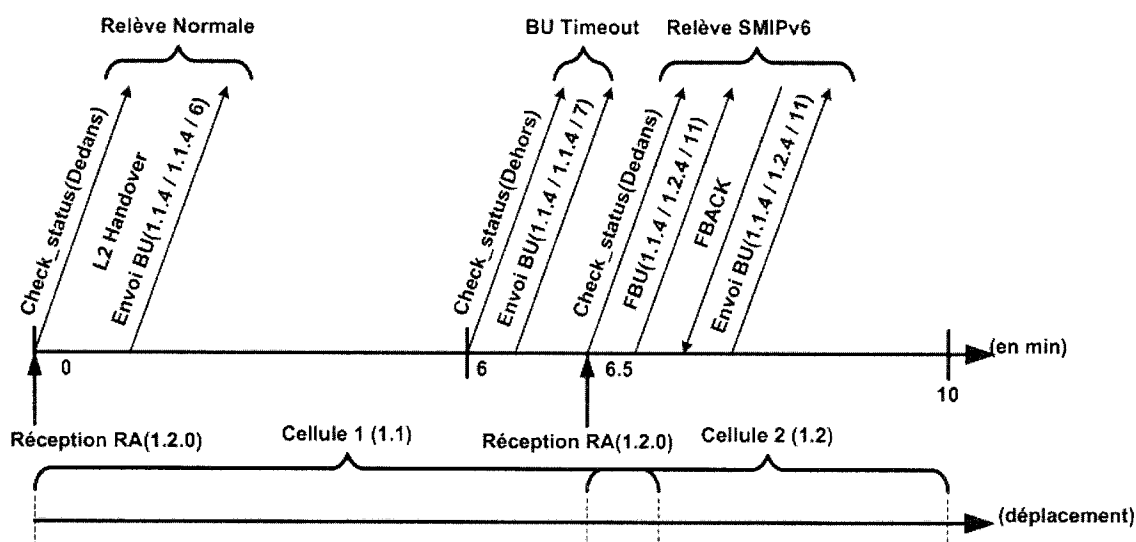
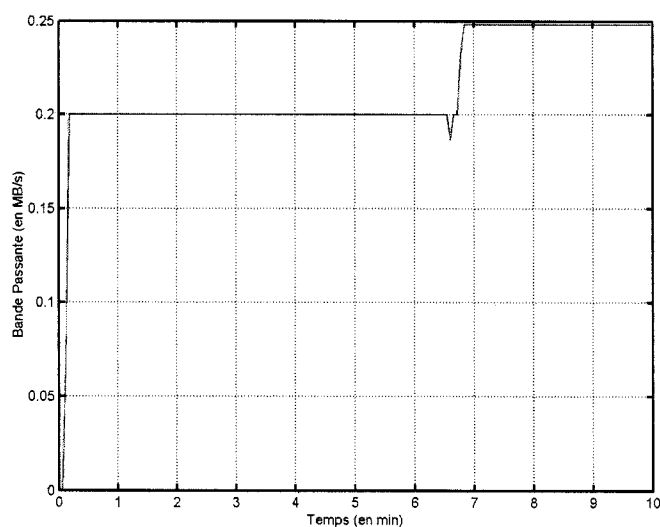


Figure 4.6 Scénario 2

Les Figures 4.7 et 4.8 présentent la variation de la bande passante en fonction du temps pour les protocoles SMIPv6 et MIPv6 respectivement. Nous remarquons que la relève, dans le cas de SMIPv6, s'est effectuée avec une stabilité de la bande passante contrairement à MIPv6 qui a encore subi une baisse drastique de la bande passante.

**Figure 4.7 Bande passante en fonction du temps – SMIPv6**

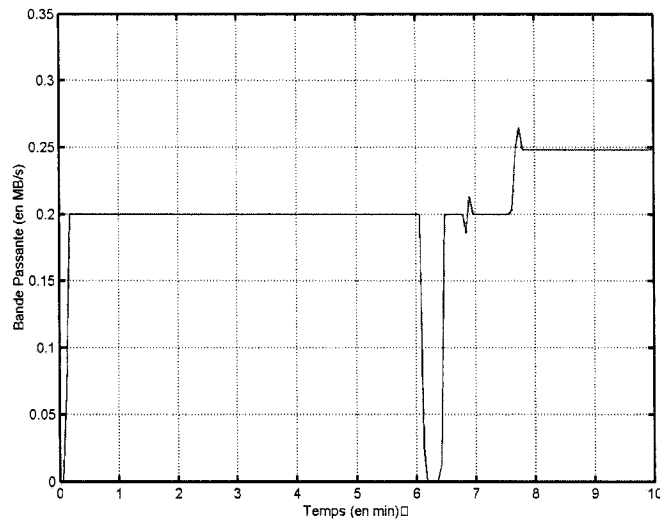


Figure 4.8 Bande passante en fonction du temps – MIPv6

Scénario 3 : Dans ce scénario, l'unité mobile ne respectera pas son profil. Cette dernière pénétrera la cellule 2 (1.2) en avance (voir Figure 4.9).

1. À 0:00, le MN vient de se mettre en marche et il détecte le signal de la cellule 1. Cette BS fait partie du profil de mobilité du MN; ainsi, le MN crée une nouvelle COA pour ce sous-réseau. Il se connecte à cette nouvelle BS et envoie un BU avec son adresse actuelle (1.1.4) et une durée de vie *current_entry.end_time + delay* (0 :06).
2. À 0:03, le MN détecte le signal de la cellule 2. Il vérifie son état et trouve que la cellule 2 ne fait pas partie de son profil. Il attend un RA pour créer une COA pour ce nouveau réseau. Il se connecte à cette nouvelle cellule et envoie un BU avec son adresse actuelle (1.2.4) et une durée de vie *max_rate* (0 :04).
3. À 0:04, le BU expirera et va entraîner une vérification de l'état du MN. À 0:04, la cellule 2 fait partie du profil; ainsi, le MN envoie un BU avec son adresse actuelle (1.2.4) et une durée de vie *current_entry.end_time + delay* (0:11).

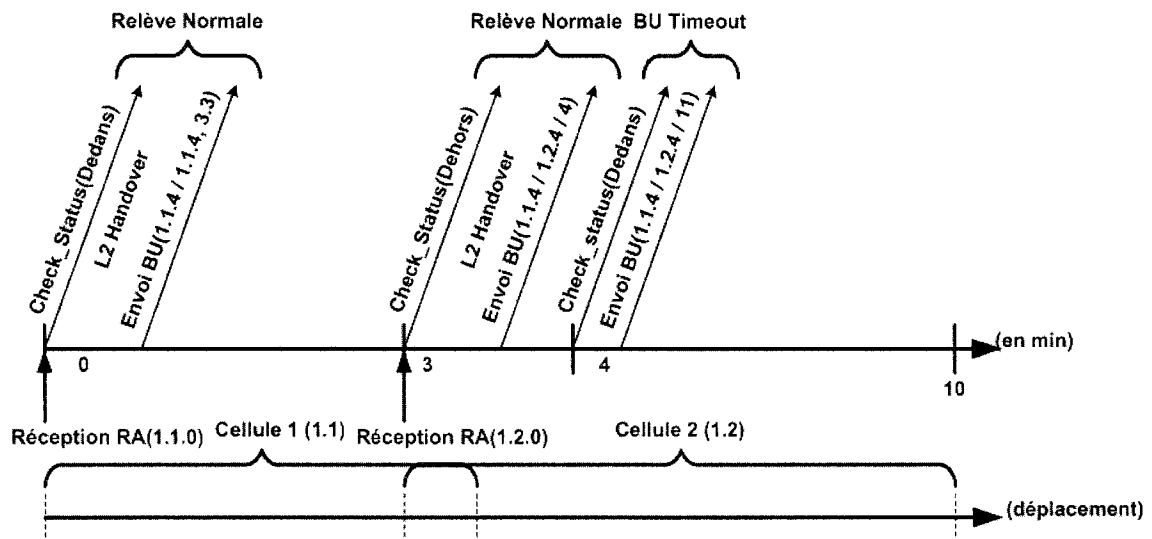


Figure 4.9 Scénario 3

Les Figures 4.10 et 4.11 présentent la variation de la bande passante en fonction du temps pour les protocoles SMIPv6 et MIPv6 respectivement. Nous pouvons remarquer que la relève normale dans le cas de SMIPv6 à 0:03 se produit avec la même stabilité que la relève SMIPv6 effectuée dans le scénario 1. MIPv6 effectue une relève avec une baisse important de la bande passante.

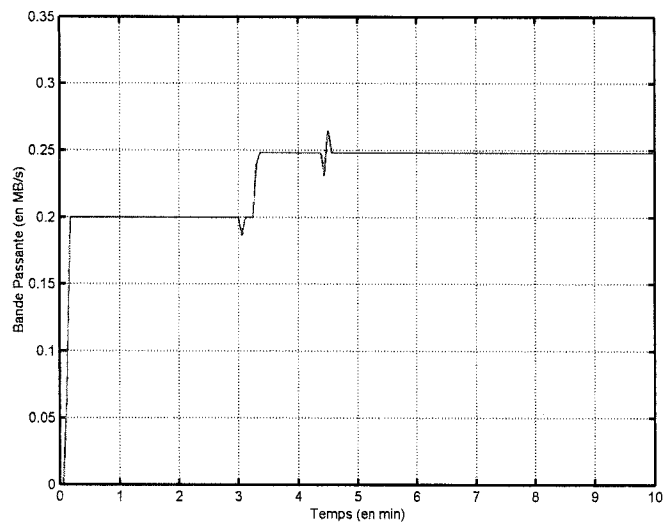


Figure 4.10 Bande passante en fonction du temps – SMIPv6

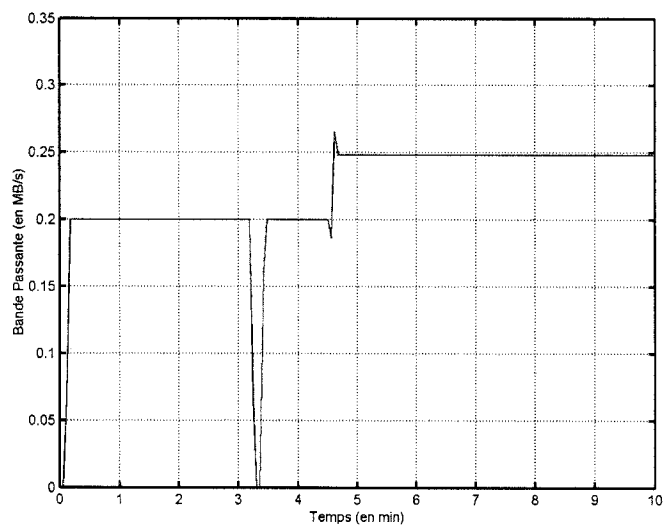


Figure 4.11 Bande passante en fonction du temps – SMIPv6

4.3 Plan d'expérience

Cette section commence par élaborer les métriques utilisées pour l'évaluation de performance. Elle présente la topologie et les conditions du réseau servant de base au plan d'expérience.

Pour évaluer la performance du modèle de gestion de mobilité proposé nous utiliserons quatre métriques de performance, que nous comparerons pour chaque utilisation modélisée :

- le délai de la relève mesure la période durant laquelle l'unité mobile ne peut ni recevoir ni envoyer de paquets IP durant une relève ;
- la perte de paquets IP permet de quantifier les transmissions qui échouent durant une relève ;
- la quantité de signalisation compte le nombre de paquets échangés pour effectuer une relève ;
- la bande passante mesure la quantité de paquets de données reçus pendant la simulation.

La topologie présentée a été conçue de manière à être suffisamment grande pour fournir des résultats réalistes tout en étant suffisamment petite pour être simulée sur NS-2. La topologie choisie, illustrée à la Figure 4.12, est composée par un *Home Agent* (HA) et un *Correspondent Node* (CN) qui sont connectés à travers l'Internet (représenté par le lien ajustable (ld) à un routeur central (CR). Six routeurs d'accès (AR), chacun représentant un sous-réseau IP différent, sont connectés à travers 3 routeurs intermédiaires (IR) au routeur central. Les routeurs d'accès ont été positionnés de manière à procurer une zone de couverture totale de 1000x3000 mètres carrés, vu que le rayon de transmission est de 250 mètres. La technologie d'accès utilisé est le 2 Mbps WLAN 802.11 fourni par NS-2.

Les liens qui connectent les AR, IR et CR sont modélisés par des liens de 5 Mbps duplex avec un délai de transmission de 2 ms. Le réseau Internet qui connecte le CR aux HA et CN est modélisé par un lien de 5 Mbps duplex avec une valeur par défaut du délai de transmission (ld) de 10 ms. Dans les simulations, la valeur de ld sera variée pour représenter différentes distances entre le MN et les HA/CN.

Pour simuler un trafic en temps réel, nous créerons un nœud correspondant (CN) comme étant une source de trafic constant (CBR) en utilisant le protocole UDP. Les paquets auront une taille de 200 octets. Un correspondant qui génère un trafic CBR peut être assimilé à un trafic audio ou *Voice over Internet Protocol* (VOIP).

Les facteurs qui semblent influencer appréciablement la valeur des indices de performance définis auparavant sont : le délai de transmission sur Internet (ld), le débit de paquets provenant du CN ($pckt_rate$), la vitesse du MN (mn_speed) et la zone de chevauchement ($intersec$).

Le délai de transmission sur Internet (ld) est un facteur important vu que le délai de mise à jour des HA et CN dépend beaucoup de cette valeur. Nous choisirons les valeurs 2, 10, 30, 50, 70, 90 et 100 ms pour pouvoir représenter une mobilité locale et globale du MN. Le débit de paquets provenant du CN ($pckt_rate$) est un facteur important, vu que le nombre de paquets perdus durant la relève en dépend énormément. Nous choisirons des débits de 4, 10, 25, 50, 75 et 100 paquets/sec pour pouvoir couvrir tous les genres de trafics UDP. La vitesse du MN est un facteur pertinent vu que le nombre de paquets perdus pendant la relève en dépend énormément, ainsi que la possibilité d'échec d'une relève d'ailleurs (perte de signalisation utilisée pour établir une relève). Nous choisirons les vitesses : 100 km/h (vitesse d'une voiture), 300 km/h (vitesse d'un train à grande vitesse) et plusieurs vitesses intermédiaires. Par ailleurs, la zone de chevauchement entre deux cellules représente un bon facteur à étudier, vu que la taille de cette zone affecte directement le temps disponible pour effectuer une relève. Nous choisirons les valeurs 5, 10 et 15 m.

Les niveaux de ces facteurs seront choisis de manière à couvrir adéquatement l'étendue de variation de ce facteur. D'un autre côté, nous minimiserons le nombre total de niveaux pour des raisons économiques. La Tableau 4.1 représente les différents niveaux des facteurs :

Tableau 4.1 Niveaux des facteurs

Nom	Symbole	Niveaux	Unité
Délai de la transmission Internet	ld	2, 10, 30, 50, 70, 90, 100	ms
Débit de paquets	pckt_rate	10, 25, 50, 75, 100	paquets/s
Vitesse du MN	mn_speed	10, 50, 100, 150, 200, 250, 300	km/h
Zone de chevauchement	intersec	5, 10, 15	m

Nous ferons un plan d'expérience un facteur à la fois pour mieux remarquer l'influence de chaque facteur sur les indices de performance observés et pour minimiser le nombre d'expériences comparé à une conception factorielle. Vu que nous n'implémentons pas un module d'apprentissage de la mobilité, nous allons tester notre protocole sur différents types d'utilisateurs mobiles. En effet, nous considérons 4 genres d'utilisateurs :

- déterministes dont la mobilité peut être prédite à 100%;
- quasi-déterministe dont la mobilité peut être prédite à 90% ;
- quasi-aléatoire dont la mobilité peut être prédite à 70% ;
- aléatoire dont la mobilité peut être prédite à 50%.

Ainsi, nous considérons dans chaque simulation les 5 cas suivants : protocole MIPv6, protocole SMIPv6 où l'utilisateur suit son profil à 100%, 90%, 70% et 50%.

4.4 Analyse des résultats

Dans cette section, nous présentons les résultats de simulation de la comparaison entre *Mobile IPv6* et *Seamless Mobile IPv6*. Comme décrit dans le plan d'expérience, nous allons tester notre protocole sur différents types d'utilisateurs mobiles : déterministe, quasi-déterministe, quasi-aléatoire et aléatoire. Les Tableaux A.1 à A.4 présentent les résultats des simulations du protocole SMIPv6 avec un utilisateur mobile qui suit son profil à 100%, 90%, 70% et 50% (voir annexe). Le Tableau A.5 représente les résultats des simulations du protocole MIPv6 (voir annexe). Nous allons analyser les résultats des simulations des deux protocoles selon les facteurs choisis : le délai de transmission sur

Internet en ms (ld), le débit de paquets provenant du CN exprimé par l'intervalle entre les paquets en ms ($pckt_rate$), la vitesse du MN en m/s (mn_speed) et la zone de chevauchement en m ($intersec$).

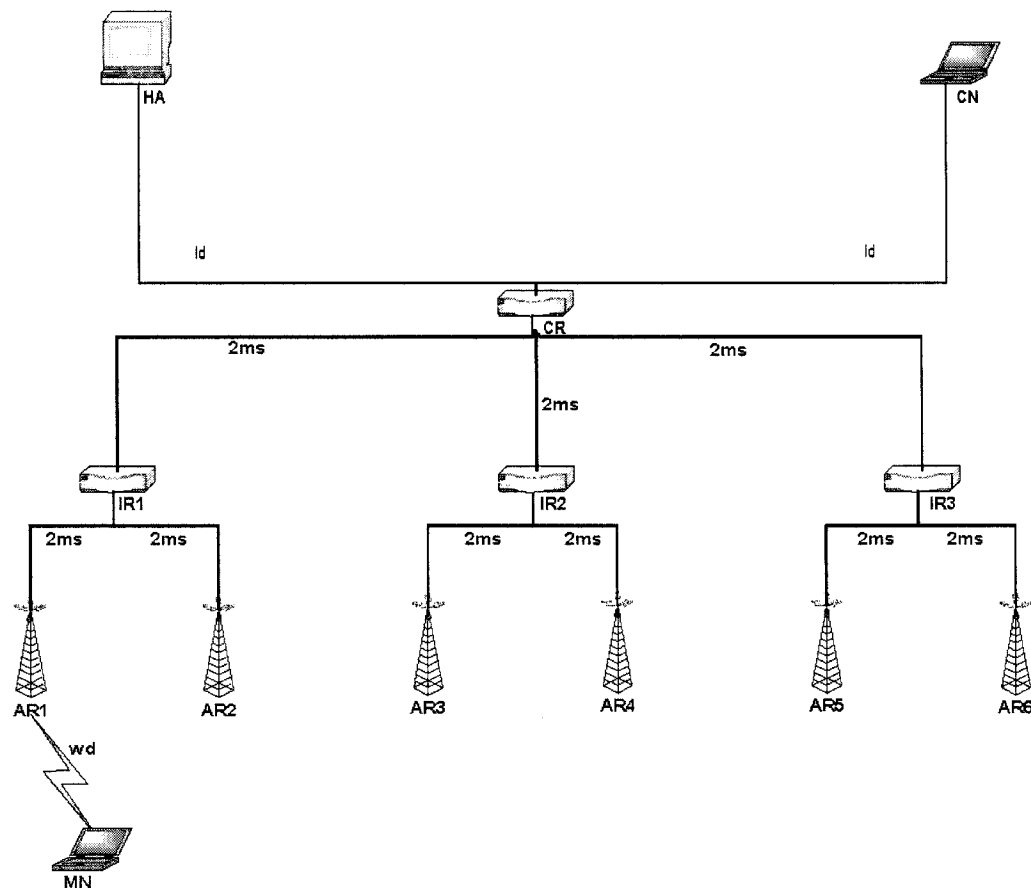


Figure 4.12 Topologie du réseau

Délai de la transmission Internet

La Figure 4.13 représente le délai de la relève en fonction du délai de la transmission sur Internet. Les autres facteurs ont les valeurs suivantes : $pckt_rate = 4$ ms, $speed = 27.8$ m/s, $intersec = 5$ m.

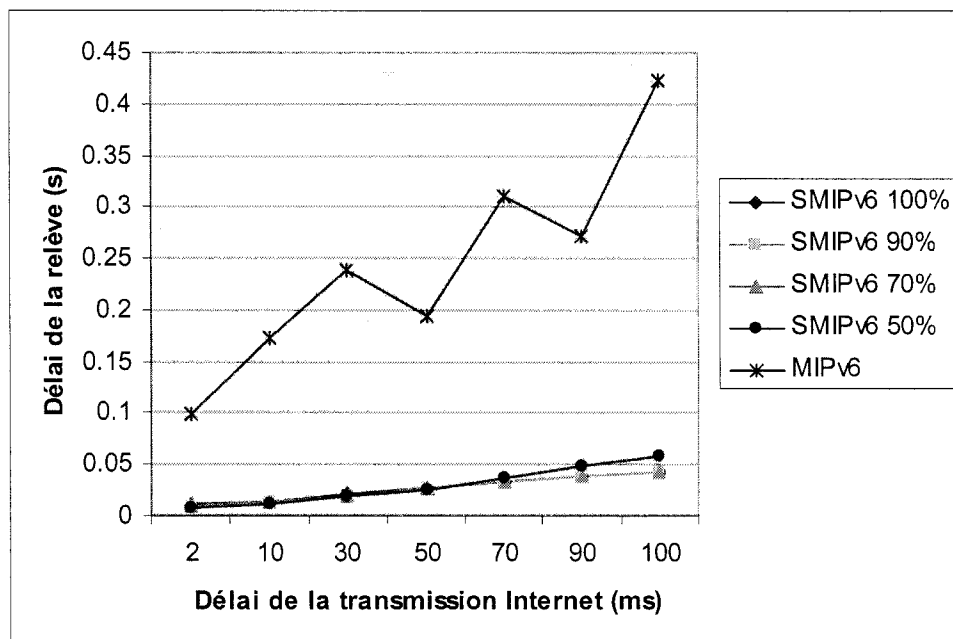


Figure 4.13 Délai de la relève en fonction du délai de la transmission sur Internet

Nous constatons que, indépendamment de la régularité de l'utilisateur considéré, le protocole SMIPv6 propose de plus bas délais de relève. En effet, la courbe représentant le délai de la relève pour le protocole MIPv6 est largement au dessus des courbes représentant le délai de la relève pour le protocole SMIPv6. D'après l'analyse mathématique, le protocole SMIPv6 a un délai de relève de $4f + 4T_2$ où T_2 désigne le délai de traitement d'un nœud et f le délai d'un lien filaire. A priori, le délai de la relève serait constant pour le protocole SMIPv6. Cependant, le trajet de notre usager traverse à plusieurs reprises son réseau d'origine où le délai des liens filaires a été modifié à l_d . Ainsi, dans ce scénario là, le délai de la relève augmente avec la valeur de l_d . Malgré ce comportement indésirable, le protocole SMIPv6 offre un délai de relève beaucoup plus bas que MIPv6.

La Figure 4.14 représente le nombre de paquets perdus en fonction du délai de la transmission sur Internet. Les autres facteurs ayant les valeurs suivantes : $\text{pkt_rate} = 4 \text{ ms}$, $\text{speed} = 27.8 \text{ m/s}$, $\text{intersec} = 5 \text{ m}$.

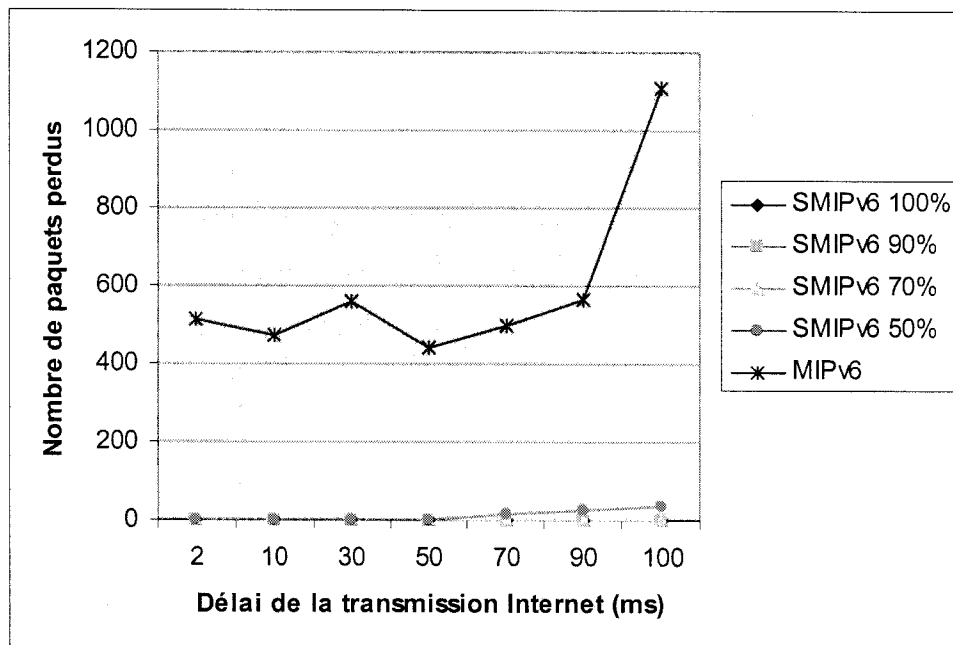


Figure 4.14 Nombre de paquets perdus en fonction du délai de la transmission sur Internet

Nous constatons que le protocole SMIPv6 ne subit aucune perte de paquets durant ses relèves, sauf dans le cas où l'utilisateur suit son profil à 50%. Cependant, MIPv6 subit des pertes de paquets importantes. En effet, une relève dans le protocole MIPv6 n'a aucun système de redirectionnement des paquets durant la relève, contrairement à SMIPv6. SMIPv6 élimine la perte de paquets durant ses relèves en anticipant un redirectionnement de paquets.

La Figure 4.15 représente le nombre de BU envoyés en fonction du délai de la transmission sur Internet. Les autres facteurs ayant les valeurs suivantes : $\text{pkt_rate} = 4 \text{ ms}$, $\text{speed} = 27.8 \text{ m/s}$, $\text{intersec} = 5 \text{ m}$.

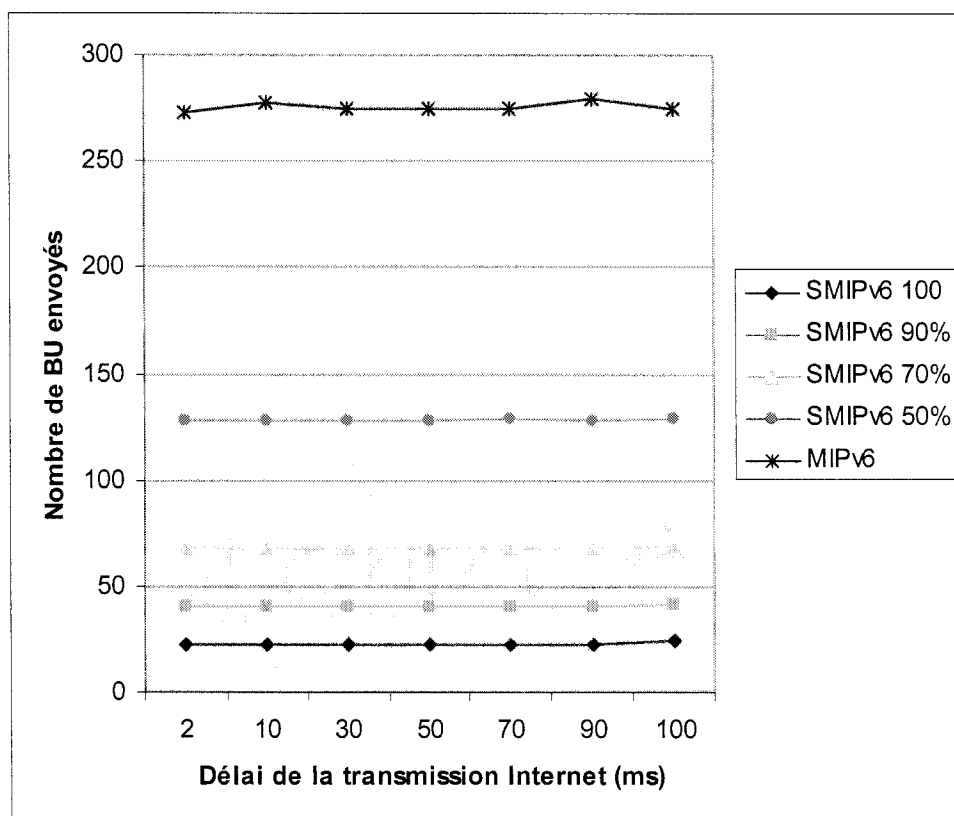


Figure 4.15 Signalisation en fonction du délai de la transmission sur Internet

Nous constatons que le nombre de BU ne dépend pas du délai de la transmission sur Internet. En effet, le nombre de BU envoyés dépend seulement de la trajectoire de l'utilisateur. Selon la Figure 4.15, le nombre de BU envoyés croît avec le manque de régularité de l'utilisateur. Un usager mobile qui quitte son profil commencera à envoyer des BU à un intervalle de temps défini au lieu de n'envoyer un BU qui au moment de changer de cellule. Selon la Figure 4.15, le protocole SMIPv6 génère beaucoup moins de signalisation par le nombre de BU envoyés.

La Figure 4.16 représente la bande passante en fonction du délai de transmission sur Internet. Les autres facteurs ont les valeurs suivantes : $\text{pkt_rate} = 4 \text{ ms}$, $\text{speed} = 27.8 \text{ m/s}$, $\text{intersec} = 5 \text{ m}$.

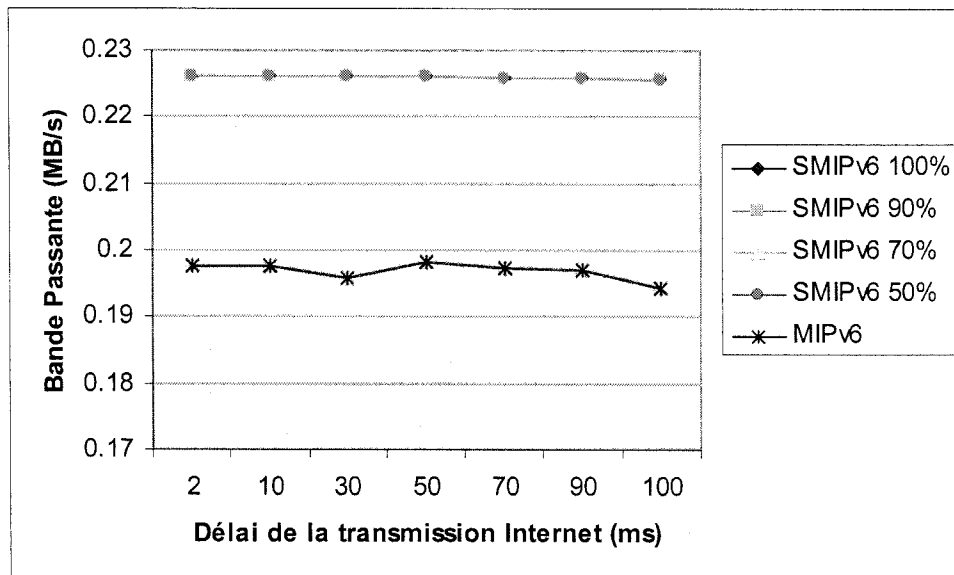


Figure 4.16 Signalisation en fonction du délai de la transmission sur Internet

Nous constatons que la bande passante dans le cas MIPv6 est très basse, comparé au protocole SMIPv6. En effet, la bande passante est plus basse vu la quantité importante de paquets perdus durant les relèves. Ainsi, SMIPv6 permet de maintenir une bande passante beaucoup plus élevée que dans le cas de MIPv6.

Débit de paquets

La Figure 4.17 représente le nombre de paquets perdus en fonction du débit de paquets. Les autres facteurs ont les valeurs suivantes : $l_d = 100$ ms, $speed = 27.8$ m/s, $intersec = 5$ m.

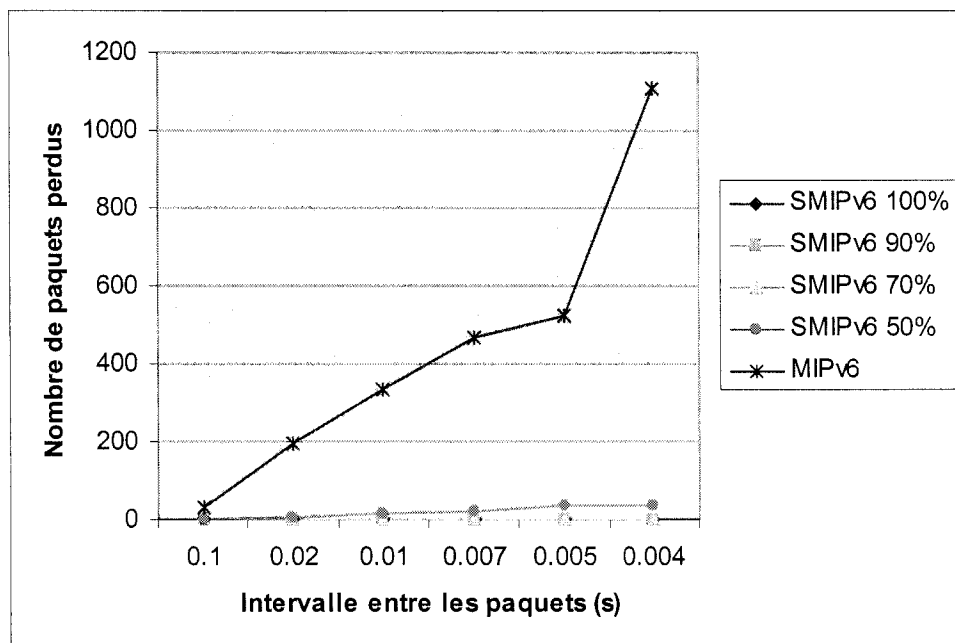


Figure 4.17 Nombre de paquets perdus en fonction du débit de paquets

Nous constatons que le protocole SMIPv6 ne subit aucune perte de paquets durant ses relèves, sauf dans le cas où l'utilisateur suit son profil à 50%. Cependant, MIPv6 subit des pertes de paquets importantes. En effet, la perte de paquets croît avec l'augmentation du débit vu que le nombre de paquets émis pendant la relève est plus important.

Zone de chevauchement

La Figure 4.18 représente le nombre de paquets perdus en fonction du débit de paquets. Les autres facteurs ont les valeurs suivantes : $l_d = 100$ ms, $pkt_rate = 4$ ms, $speed = 27.8$ m/s.

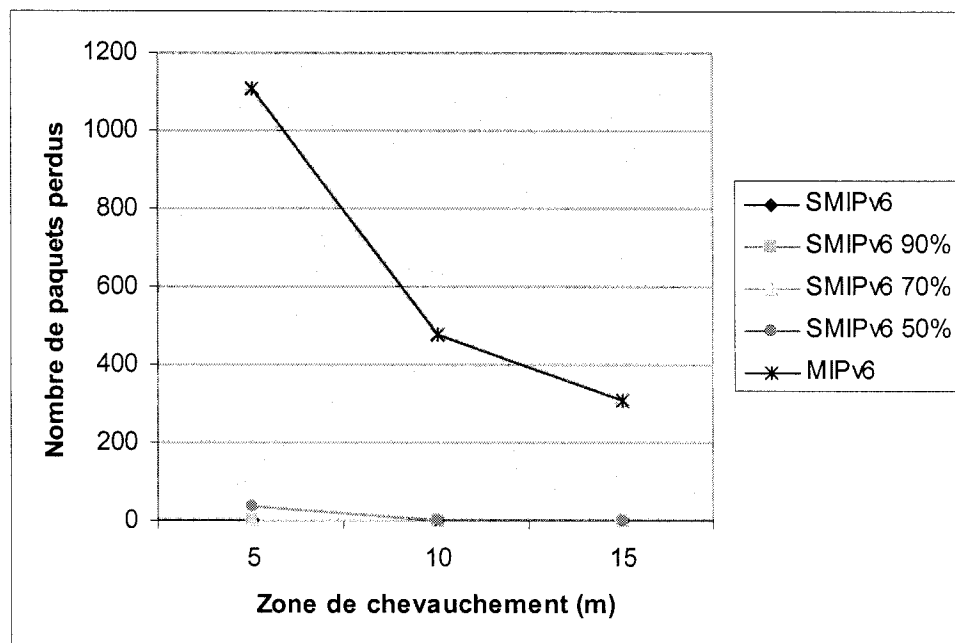


Figure 4.18 Nombre de paquets perdus en fonction de la zone de chevauchement

Nous constatons que le protocole SMIPv6 n'est pas très influencé par la taille de la zone de chevauchement. En effet, nous avons décidé de faire varier ce facteur seulement pour vérifier si notre protocole serait assez rapide pour effectuer une relève même avec des zones de chevauchement relativement petites et à des vitesses importantes. Nous remarquons ainsi que notre protocole est assez rapide pour prédire une relève et effectuer les échanges nécessaires. En effet, le protocole MIPv6 subit des pertes d'autant plus importantes que la zone de chevauchement est petite. Or SMIPv6 ne subit des pertes que dans le cas de l'utilisateur aléatoire avec une zone de chevauchement de 5m.

La Figure 4.19 représente la bande passante en fonction du débit de paquets. Les autres facteurs ont les valeurs suivantes : $l_d = 100$ ms, $pckt_rate = 4$ ms, $speed = 27.8$ m/s.

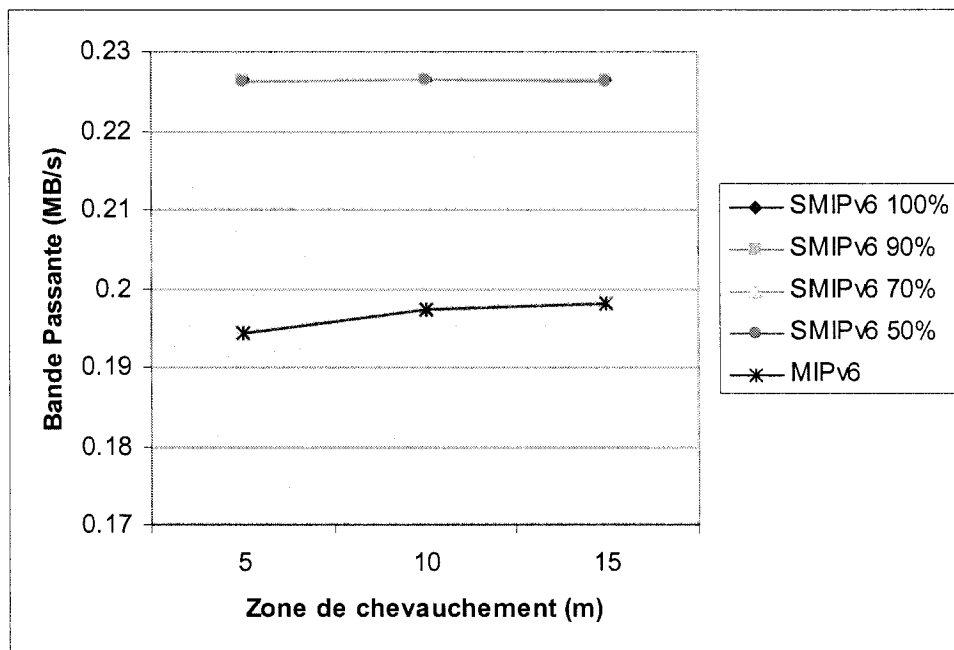


Figure 4.19 Bande passante en fonction de la zone de chevauchement

Nous constatons que le protocole SMIPv6 n'est pas très influencé par la taille de la zone de chevauchement. En effet, nous avons décidé de faire varier ce facteur seulement pour vérifier si notre protocole serait assez rapide pour effectuer une relève même avec des zones de chevauchement relativement petites et à des vitesses importantes. Nous remarquons ainsi que notre protocole maintient une bande passante constante.

Vitesse du MN

La Figure 4.20 présente le délai de la relève en fonction de la vitesse du MN. Les autres facteurs ont les valeurs suivantes : $ld = 100$ ms, $pckt_rate = 4$ ms, $intersec = 5$ m.

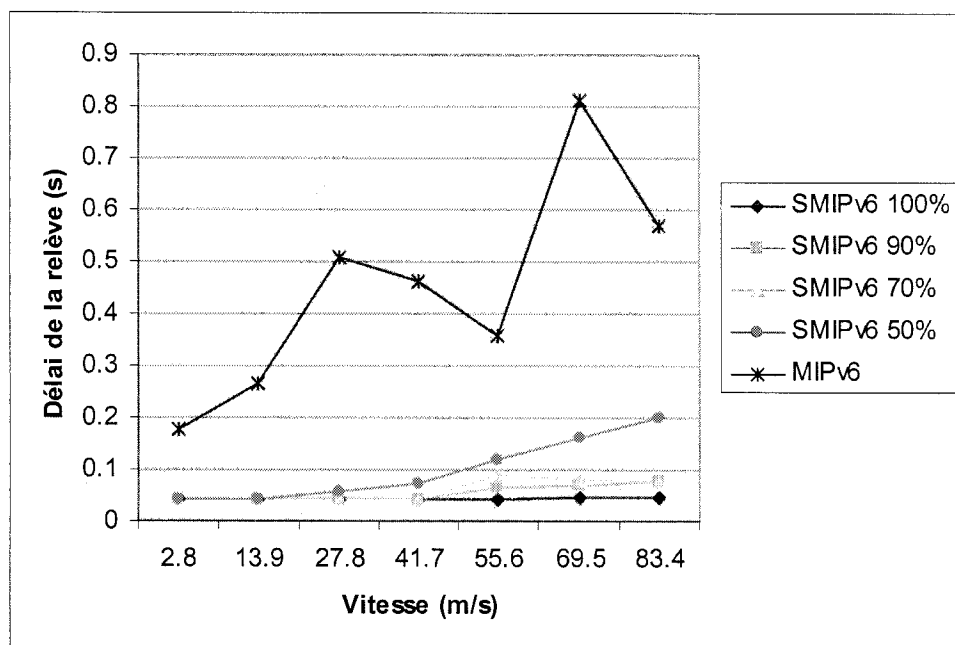


Figure 4.20 Délai de la relève en fonction de la vitesse du MN

Nous constatons que le délai de la relève du protocole SMIPv6 reste toujours largement plus bas que dans le cas de MIPv6. Les délais de la relève dans MIPv6 n'augmentent pas proportionnellement à la vitesse, vu que ces derniers se basent sur le moment où le premier RA de la nouvelle cellule est reçu pour enclencher une relève.

La Figure 4.21 représente la perte de paquets en fonction de la vitesse du MN. Les autres facteurs ont les valeurs suivantes : $l_d = 100$ ms, $pckt_rate = 4$ ms, $intersec = 5$ m.

Nous constatons que la courbe représentant le nombre de paquets perdus pour le protocole MIPv6 est au-dessus de celles pour le protocole SMIPv6. En effet, le protocole MIPv6 subit d'importantes pertes de paquets pendant les simulations. La courbe représentant le nombre de paquets perdus pour le protocole MIPv6 a la même allure que celle représentant le délai de la relève en fonction de la vitesse pour le protocole MIPv6. Ces deux métriques sont très liées vu que le nombre de paquets perdus correspond au produit du délai de la relève et du débit de paquets. Dans le cas de MIPv6, ces deux

métriques sont très reliées vu que MIPv6 n'a pas un système de redirectionnement des paquets qui permet d'éviter la perte de paquets, contrairement à SMIPv6.

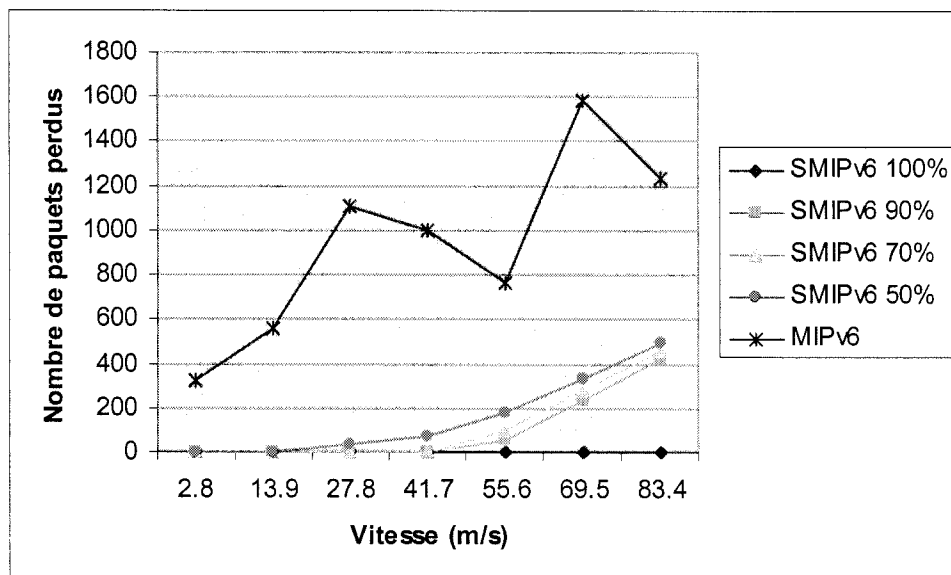


Figure 4.21 Paquets perdus en fonction de la vitesse du MN

CHAPITRE V

CONCLUSION

Le présent mémoire a porté sur le développement d'un nouveau modèle de gestion de la mobilité permettant une meilleure relève intercellulaire. Nous effectuons une synthèse de nos travaux dans la première section de ce chapitre. La deuxième section énonce les limitations relatives à nos méthodes ainsi qu'à leur implémentation. Enfin, nous terminons ce chapitre en énonçant dans une troisième section les travaux futurs qu'il serait possible d'entreprendre afin de poursuivre les efforts entrepris dans cette recherche, dans le but de les appliquer éventuellement à d'autres problèmes similaires.

5.1 Synthèse des travaux

Dans ce mémoire, nous avons proposé un nouveau modèle de gestion de mobilité pour les réseaux IPv6 se basant sur les patrons de mobilité des usagers, appelé *Seamless Mobile IPv6*. Tout d'abord, nous avons présenté le protocole Mobile IPv6, en précisant les aspects de la mobilité dans l'Internet. Ensuite, nous avons posé les éléments de la problématique de la mobilité dans les réseaux IPv6. Par la suite, nous avons présenté des protocoles de mobilité complémentaires à Mobile IPv6 recensés dans la littérature sous l'angle de leur efficacité et leur performance.

En partant de ces analyses, nous avons développé un algorithme de gestion de mobilité plus complet et plus efficace. En effet, nous avons proposé un modèle qui vise à réduire les coûts de la signalisation engendrée par la mise à jour des correspondants, ainsi que les délais de relève entre les réseaux lors des déplacements de l'unité mobile. Le modèle proposé exploite l'opportunité qu'offre l'utilisation des profils de mobilité des usagers pour permettre une relève sans coupure. En effet, *Seamless Mobile IPv6* se

base sur les profils de mobilité des usagers afin de prédire la prochaine cellule où la relève va se produire afin de mettre à jour ses correspondants avant de quitter son réseau courant. Le moment de la relève peut être perdu grâce à des informations provenant du réseau. D'un autre côté, l'usage du profil de mobilité va permettre d'effectuer une mise à jour de sa localisation auprès de ses correspondants une fois un changement de réseau est en vue ce qui réduit les coûts de la signalisation. Par ailleurs, nous avons présenté le module d'apprentissage de la mobilité des usagers utilisé pour générer les patrons de mobilité. Ensuite, nous avons introduit un modèle analytique pour étudier la performance de notre modèle comparativement à d'autres stratégies de gestion de mobilité. Par la suite, nous avons illustré le fonctionnement de notre modèle à travers plusieurs scénarios. Nous avons mené plusieurs simulations pour analyser les comportements de ce nouveau modèle. Les résultats numériques obtenus ont montré qu'une réduction significative du délai de la relève est obtenue en utilisant le protocole *Seamless Mobile IPv6*. Ces résultats ont aussi montré qu'une réduction importante sur les coûts de la signalisation a été obtenue en utilisant le protocole proposé.

5.2 Limitations des travaux

Comme dans tout travail de recherche, il y a toujours une place à l'amélioration. Le protocole proposé pour la gestion de la mobilité permet d'améliorer la mobilité dans les réseaux IPv6. Notre protocole a démontré d'excellentes performances même dans le cas d'une mobilité aléatoire. Cependant, notre protocole se base sur un module d'apprentissage de la mobilité que nous n'avons pas implémenté et dont nous ne connaissons guère les performances. D'un autre côté, le protocole proposé utilise entre autres certains aspects de la relève du protocole *Fast Handovers for Mobile IPv6* (FMIPv6) qui est un protocole en voie de normalisation. En effet, cette méthode de prédiction de la relève se base sur une assumption importante qui est le fait d'avoir une information de la couche 2 du réseau nous avisant d'une relève. D'un autre côté, le simulateur *Network Simulator* (NS-2), qui est un simulateur très puissant et très avancé

dans certains cas, n'était pas très facile d'usage dans notre cas. En effet, le module *MobiWan* utilisé pour simuler Mobile IPv6 a été développé sur une ancienne version de NS et n'a pas été mis à jour depuis. Aussi, il n'y a pas tellement de documentation, de soutien technique ou d'aide pour l'utilisation de NS-2 ou de Mobiwan. Il ne faut pas oublier aussi l'extrême instabilité et les limitations matérielles de ce simulateur. En effet, tout cela est dû au fait que MIPv6 est un protocole en cours de normalisation d'où la rareté de simulateurs qui le supportent.

5.3 Indication de travaux futurs

Il convient de mentionner, malgré toutes les recherches effectuées et les protocoles proposés, que le problème de mobilité dans l'Internet demeure encore un problème ouvert et d'actualité. Les recherches futures devraient être orientées vers les schémas de prédiction de la mobilité, vu que la plupart des protocoles proposés sont plutôt réactifs que proactifs. Les stratégies de prédiction basées sur le patron de mobilité des usagers exigent la collecte et le traitement en temps réel pour que les patrons de mobilités puissent être ajustés dynamiquement pour chaque utilisateur. Ce genre de collecte et de traitement exige un support efficace de calcul de tous les usagers. Il est également nécessaire d'étudier l'impact entre l'utilisation d'une architecture distribuée et une architecture centralisée. En effet, une architecture distribuée engendre une étude de la limitation matérielle d'un terminal mobile pour comprendre un module d'apprentissage de sa mobilité. Il serait également intéressant d'étudier l'impact de la prédiction de la mobilité des usagers sur la planification et la modélisation des réseaux.

ANNEXE

Annexe 1 Présentation de Network Simulator NS-2

Network Simulator (NS) est un simulator pour les réseaux de communication, développé au *Lawrence Berkeley national Laboratory* (LBNL) au sein du projet VINT. Il est rapidement devenu un outil essentiel pour la communauté de chercheurs puisque tout le monde peut intégrer ses propres modules et contribuer au développement du simulateur. *Network Simulator* est gratuit et fonctionne sous Unix, Linux et Windows. La première version étant à titre expérimental, nous utilisons maintenant la deuxième version appelée NS-2. NS est principalement utilisé pour la simulation des protocoles Internet, plus particulièrement les protocoles TCP et UDP. Cependant, sa bonne organisation hiérarchique a motivé beaucoup de chercheurs à étudier :

- de nouveaux protocoles qu'ils proposent pour les différentes couches du réseau Internet (Routage, Transport et Application) ;
- l'impact de nouveaux media de transmission (ATM, Satellites, LAN et WLAN) ;
- et la performance de nouvelles architectures proposées pour améliorer la qualité de service dans Internet (DiffServ, IntServ, *buffer management*).

Annexe 1.1 Principes de base

NS est un simulateur orienté objet. Une simulation n'est plus ni moins que le déplacement de paquets, qui sont des objets, parmi différents objets représentant les éléments et protocoles du réseau. NS est écrit en C++. Ceci permet de faciliter l'ajout de nouveaux protocoles et mécanismes. Les principales classes sont :

- *Application* : Classe mère de la plupart des applications (ex. *ftp*, *telnet*) ;
- *Agent* : Classe mère de tous les protocoles qui tournent aux couches 3 et 4 (ex. TCP, UDP, TFRC, RTP, RIP, OSPF, SRM, DVMRP, PIM) ;

- *Node* : représente l'entité de nœuds dans le réseau. Un nœud peut être un terminal, un commutateur, un routeur, ou une passerelle. Chaque nœud contient un *Classifier*, jouant le rôle d'une table de routage, qui décide où envoyer un paquet qui vient d'une interface ou d'un agent. Le paquet doit être envoyé à un agent ou une interface extérieure ;
- *Queue* : Classe mère de toutes les files d'attente ou tampons (ex. *Drop Tail*, *Drop From Front*, RED) ;
- *LinkDelay* : Cette classe simule le délai de propagation. Conjointement avec *Queue*, cette classe simule les couches 1 et 2 ainsi que la gestion de la file au niveau IP ;
- *Packet* : Cette classe simule les paquets diffusés sur le réseau. Elle pointe vers deux autres classes, la première étant *Header* qui représente l'entête et la deuxième *Payload* qui représente le contenu. Pour définir un nouveau protocole, il est nécessaire de définir une nouvelle classe *Header*;
- *TimerHandler* : La classe mère de toutes les minuteries (*Timers*) utilisées par les protocoles. Quand la minuterie (*Timer*) d'un protocole expire, une fonction particulière de la classe est appelée.

Chaque objet d'une simulation possède un pointeur vers un autre objet sur le chemin vers le destinataire du paquet (ex. un objet *LinkDelay* donne un paquet à l'interface d'entrée du nœud à la sortie du lien). Pour donner un paquet à un objet, nous devons appeler la fonction *recv* de cet objet en utilisant le pointeur que nous possédons. Nous passons à cette fonction un pointeur vers ce paquet. Le pointeur de ce paquet est passé de nœud en nœud jusqu'à destination.

Une simulation est faite en définissant les objets de la simulation (tout en précisant leur paramètres), en connectant les objets entre eux (topologie et réseau) et en démarrant l'application source. Les paquets sont créés et transmis à travers le réseau. La simulation s'arrête après un certain temps ou après un certain événement.

La configuration de la simulation et des paramètres des différents objets sont passés à NS par un fichier écrit en OTCL. NS contient un interpréteur OTCL qui traduit les commandes OTCL en leur équivalent en C++.

Annexe 1.2 Le planificateur

Le simulateur NS est basé sur un planificateur (Scheduler). Son rôle est de planifier l'exécution des événements. Un événement est une action à exécuter à un temps donné dans le futur, par exemple retransmettre un paquet quand une minuterie (Timer) expire, donner un paquet à un nœud après un temps égal au délai de propagation. Ainsi, une liste d'événements est créée en commençant par les événements indiquant le début et la fin de la simulation, ensuite les objets de la simulation ajouteront d'autres événements. Ces événements sont classés dans la liste par ordre chronologique. Le planificateur exécute la liste en commençant par le début de la liste, fait avancer le temps de la simulation, exécute l'événement suivant et ainsi de suite jusqu'à l'événement indiquant la fin. Le temps dans le simulateur ne correspond pas au temps réel.

Annexe 1.3 Les nœuds dans NS

Il y a deux genres de nœuds qui nous intéressent principalement. À la Figure A.1, on distingue les nœuds fixes de type classe *Node* qui se situent dans le nuage, les nœuds mobiles dans les carrés étant de type classe *MobileNode*. Les nœuds à la frontière du nuage sont des stations de base (Base Stations) de classe *MobileNode* avec une mobilité nulle et sont utilisés pour connecter les nœuds mobiles aux nœuds fixes.

Classe *Node* : La classe *Node* a été conçue pour simuler des nœuds fixes. Les nœuds de ce type se connectent les uns aux autres par l'intermédiaire de liens appelés *Links*, qui ont des délais, bandes passantes et probabilités de perte de paquets variables. Comme proposé par SUN pour simuler Mobile IPv4, cette classe *Node* pouvait être utilisée pour simuler des nœuds mobiles. Ceci pouvait être fait en faisant une connexion et une déconnexion d'un lien à un temps spécifié (link up/link down). En effet, cette méthode ne représente pas le vrai comportement d'une relève et n'est pas efficace en

terme de simulation ou de taille de simulation vu que cela implique de créer des liens entre le nœud et tous les points d'accès possible.

Classe *MobileNode* : Les nœuds mobiles et leurs caractéristiques sans fil ont été ajoutés à NS dans le but de simuler des réseaux ad-hoc, le but étant d'évaluer les protocoles de routage ad-hoc. Contrairement aux nœuds fixes de la classe *Node*, les nœuds mobiles de la classe *MobileNode* se connectent les uns aux autres par le biais de canaux (Channels) qui est un genre de medium sans fil de type *broadcast*. Les nœuds de type *MobileNode* se déplacent dans un plan à deux dimensions d'une taille déterminée. Les nœuds mobiles peuvent communiquer avec les nœuds fixes grâce aux stations de base. Les stations de bases sont des nœuds hybrides de type *Node* et *MobileNode*. La Figure A.2 illustre la structure d'un nœud de type Mobile IPv4.

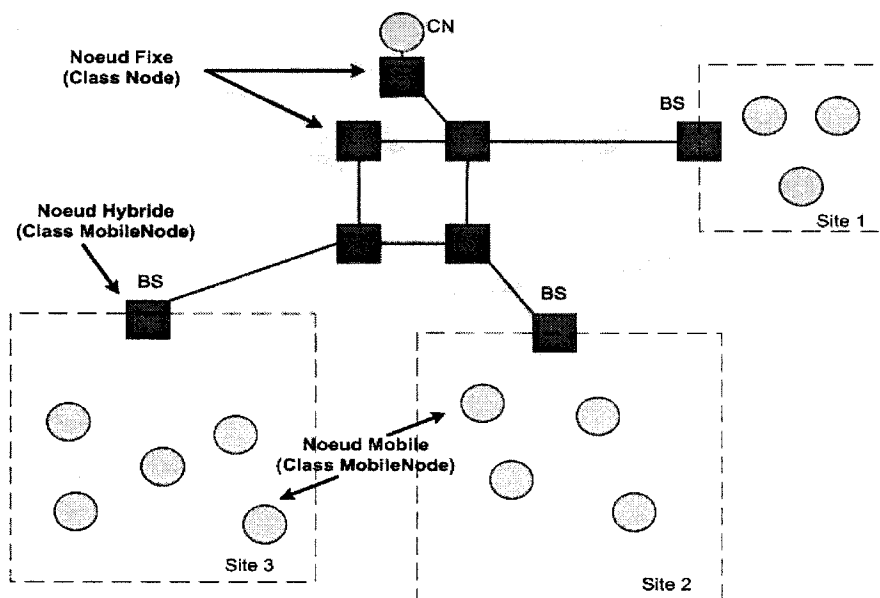


Figure A.1 Mobilité globale et locale dans NS

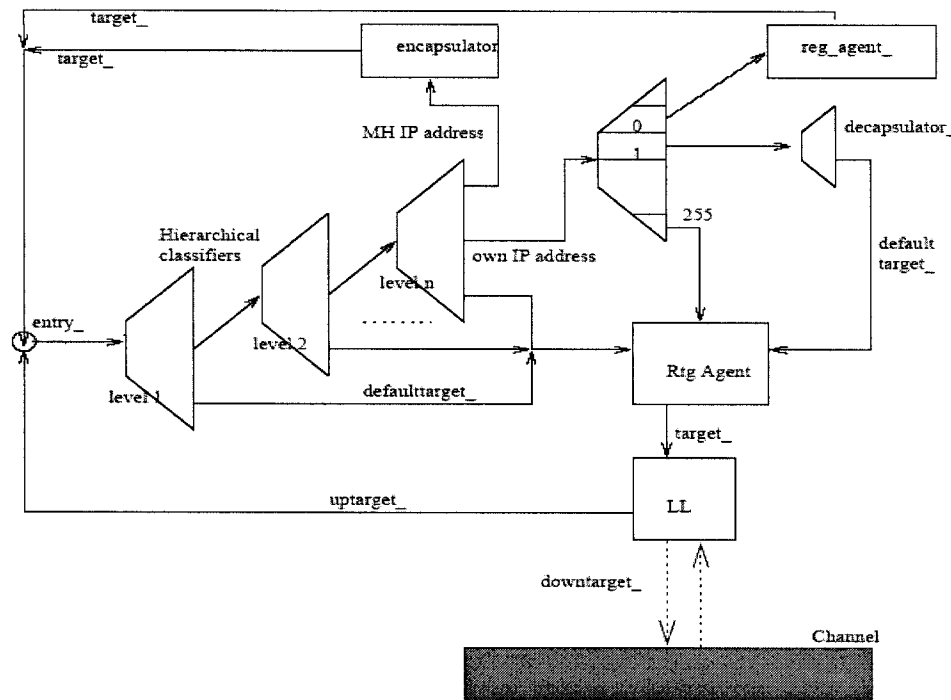


Figure A.2 Structure d'un nœud de type Mobile IPv4

Annexe 1.4 Modèles de mobilité

Le simulateur NS inclut des modèles de mobilité utilisés pour créer des scénarios de mobilité. Ces modèles sont utilisés pour les nœuds de type *MobileNode*. Les mouvements des nœuds dans le plan peuvent être spécifiés des deux manières suivantes :

- En définissant le point de départ du nœud mobile ainsi que ses déplacements futurs. Le nœud a une vitesse et se déplace vers un point spécifié. Un nouveau point de destination et une nouvelle vitesse peuvent être spécifiés à n'importe quel moment.
- Par des mouvements aléatoires des nœuds. Le nœud commence à se déplacer à une vitesse, direction et point de départ aléatoires. Une nouvelle vitesse et direction sont choisies n'importe quand dans la simulation.

Annexe 1.5 Autres utilitaires dans NS

NS comporte aussi les utilitaires suivants :

- Nam : Un utilitaire qui permet de visualiser le déplacement des paquets à travers les nœuds et les liens du réseau.
- Xgraph : Un utilitaire qui permet de dessiner les courbes de résultats de la simulation.
- Gt-item : Un outil qui permet de créer de graphes de grande taille et fortement connexes et de contrôler leur hiérarchie.
- Sgb2ns : Un utilitaire qui traduit un graphe crée par gt-item en langage OTCL pour qu'ils puissent être intégrés dans un fichier de configuration.

Annexe 2 MobiWan

MOBIWAN a été développé en Mai 2002 par Thierry Ernst au sein de MOTOROLA Paris avec la collaboration de INRIA PLANETE. MobiWan a pour but de simuler une mobilité globale des nœuds dans un réseau hiérarchisée et à grande échelle (simulations allant jusqu'à 3000 nœuds). Le module MobiWan, tournant sur la plateforme NS-2 (ns-2.1b6), comprend une extension pour simuler Mobile IPv6 ainsi que d'autres extensions pour manipuler et configurer de grandes topologies de réseaux (TOPOMAN / TOPOGEN). Ces deux extensions peuvent fonctionner indépendamment l'une de l'autre.

Annexe 2.1 Mobile IPv6 dans MobiWan

Dans cette section, nous traitons des améliorations apportées à NS-2 pour permettre le fonctionnement de Mobile IPv6. En ce qui concerne le protocole IPv6, seulement certaines fonctionnalités ont été implémentées pour simuler Mobile IPv6. Les fonctionnalités DHCPv6, *Neighbor Discovery*, etc. n'ont pas été implémentées. Principalement, le protocole Mobile IPv6 a pour but de mettre à jour les correspondants de son emplacement actuel. Par défaut, les paquets sont envoyés au réseau d'origine où ils sont interceptés par le *Home Agent* qui les encapsule et les envoie à la *Care-Of-Address* du MN. Si un CN a la *Care-Of-Address* du MN, il lui envoie le paquet

directement en utilisant l'entête de routage. Ceci nécessite l'implémentation de l'encapsulation et de la décapsulation des paquets, ainsi que le traitement de la nouvelle entête de routage IPv6 dans tous les nœuds grâce à l'utilisation de *Classifiers*.

La taille des entêtes a été modifiée. Les paquets *Router Advertisements* et *Solicitations* entre les stations de base et unités mobiles ont été rajoutés. L'encapsulation et la décapsulation des paquets à tous les nœuds ont été implémentées, ainsi que le traitement d'une nouvelle entête de routage IPv6. Le fonctionnement de Mobile IPv6 se base sur l'implémentation d'un groupe d'Agents NS tout en se basant sur les *Classifiers* NS existants. Les nœuds qui ont besoin des fonctionnalités de Mobile IPv6 contiendront des objets pour l'encapsulation, la décapsulation et pour traiter la nouvelle entête de routage IPv6. Evidemment, ceci nécessite un changement dans la manière de configurer les nœuds.

En ce qui concerne les MN, BS ou HA, la classe *MobileNode* (voir A.1.3) développée par *Rice Monarch University* a été utilisée. L'agent de routage Ad-hoc a été remplacé par un *Network Agent* (voir A.1.3). En ce qui concerne les CN, nous utilisons la classe *Node*. La structure des nœuds Mobile IPv6 est montrée à la Figure A.3 et la Figure A.4.

Annexe 2.2 Network Agent

Cet agent est utilisé par les BS et MN et remplace l'agent de routage Ad-hoc dans la classe *MobileNode*. Il est utilisé pour déterminer le prochain nœud à qui acheminer le paquet et l'inscrit dans l'entête du paquet, vérifier le temps d'expiration du paquet (TTL) et le donner au MIPv6Agent qui se charge de le diffuser en mode *broadcast*.

En ce qui concerne le MN, l'agent *NetworkMN* permet aux paquets envoyés à la *Care-Of-Address* d'être expédiés au port de démultiplexage (*demux*). Il surveille aussi les paquets entrants et informe le MNAgent de la source pour maintenir une liste de correspondant (Binding Update List).

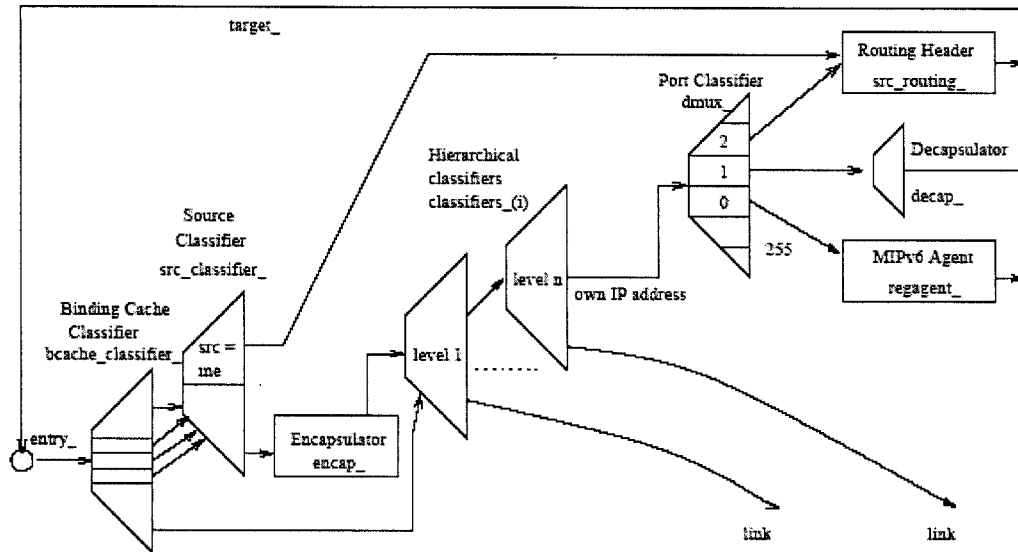


Figure A.3 Structure d'un nœud fixe Mobile IPv6 (Class Node)

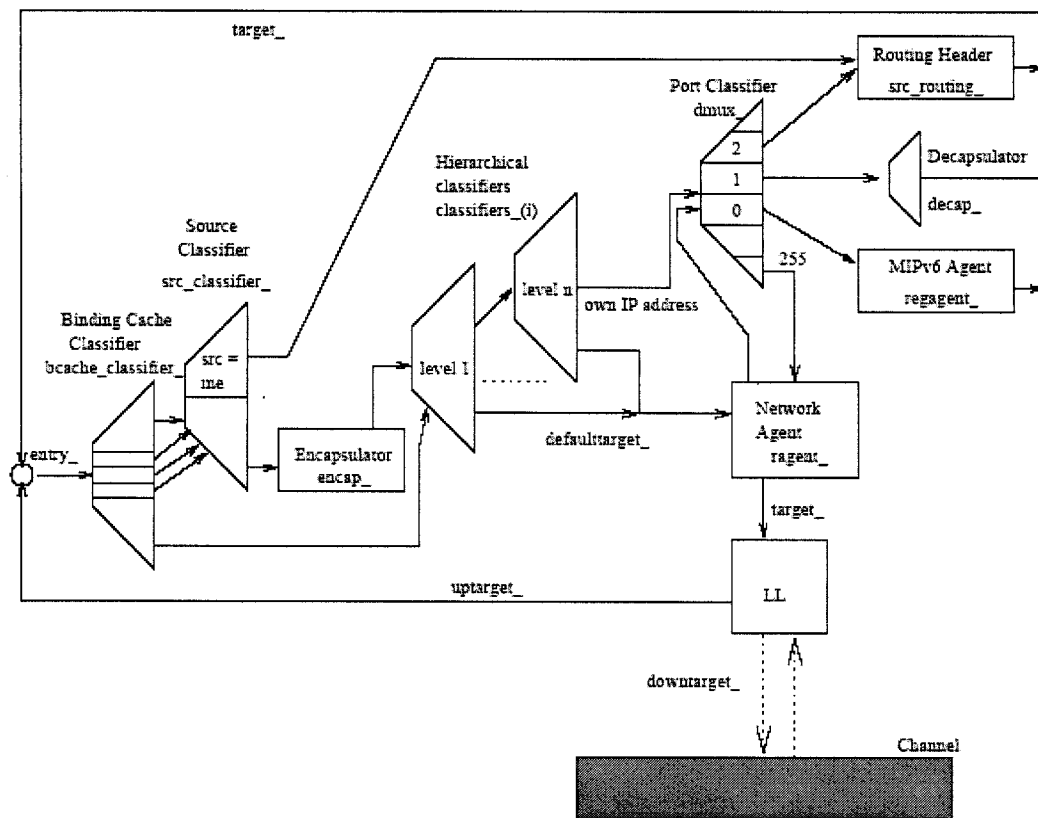


Figure A.4 Structure d'un nœud mobile MIPv6 (Class MobileNode)

Annexe 2.3 Agents Mobile IPv6

Le protocole Mobile IPv6 est implémenté comme un groupe d'agents NS, la classe de base étant MIPv6Agent de laquelle sont dérivées les classes MNAgent, CNAgent et BSAgent. Les agents Mobile IPv6 utilisent les fonctionnalités existantes :

- **Router Advertisements et Solicitations** sont directement traités dans l'agent MIPv6.
- **Base Station list** est une liste qui contient toutes les BS dont le MN a reçu un Router Advertisement. Les entrées dans cette liste sont supprimées après que le temps validité fut expiré. Quand le MN reçoit un RA provenant d'une station de base inconnue, il l'enregistre dans la liste. À ce moment là, le MN crée une nouvelle COA et utilise cette BS comme sa station de base pour communiquer.
- **Configurer une Care-Of-Address** : Les adresses sont composées par deux composants : le *network_id* qui identifie le sous-réseau et son emplacement dans la topologie, et le *node_id* qui identifie le nœud en question. L'objet ARP a été modifié pour répondre aux requêtes ARP de la COA du MN.
- **Binding Update List** : Cette liste maintenue par le MN a pour but de maintenir une liste de correspondants du MN et de maintenir les informations concernant les BU envoyés. Une entrée dans cette liste indique l'adresse du destinataire, la *Care-Of-Address*, le temps d'envoi, le numéro de séquence et la durée de vie du dernier BU envoyé, ainsi que d'autres informations pour distinguer les différents genres de nœuds (BS, HA et CN) :
 - Utilisée comme une liste de correspondants, elle contient la liste des nœuds potentiels auxquels un BU doit être envoyé. Ces nœuds sont ajoutés automatiquement par le NetworkAgent, ou explicitement par l'utilisateur par le biais de l'interpréteur OTCL dans le script définissant la simulation.
 - Utilisée comme une Binding Updates List, elle indique les nœuds auxquels les BU ont vraiment été envoyés. Les entrées dans la liste peuvent être

activées ou désactivées. Si l'entrée est active et le *Routing Optimization* est activé, des BU seront envoyés au HA et CN. Une entrée garde les informations sur le dernier BU envoyé à un certain nœud.

- **Routing Optimization** : Agent/MN met la variable *rt_opti_* à 0 pour éviter que le MN mette à jour ses correspondants. Si *rt_opti_* à la valeur 1, les BU seront automatiquement envoyés aux CN.
- **Envoi des Binding Updates** : Un MN enverra des BU quand il aura obtenu une nouvelle COA ou quand le BU expirera. Le MN envoie des BU à tous les nœuds inscrits et activés dans sa liste *Binding Update List*.
- **Binding Cache** : Quand un nœud reçoit un *Binding Update*, une entrée est ajoutée dans le *Binding Cache*. Ainsi, la table de routage doit être mise à jour pour pouvoir redirectionner les paquets à la bonne *Care-Of-Address*. Dans NS, une table de routage est implémentée sous forme d'un *Classifier*.

Annexe 3 Résultats des simulations

Les Tableaux A.1 à A.4 présentent les résultats des simulations du protocole SMIPv6 avec un usager mobile qui suit son profil à 100%, 90%, 70% et 50%. Le Tableau A.5 représentent les résultats des simulations du protocole MIPv6. Les quatre premières colonnes du tableaux représentent les facteurs que nous faisons varier : le délai de transmission sur Internet en ms (*ld*), le débit de paquets provenant du CN exprimé par l'intervalle entre les paquets en ms (*pckt_rate*), la vitesse du MN en m/s (*mn_speed*) et la zone de chevauchement en m (*intersec*). Les quatre dernières représentent les métriques de performance : le délai de la relève en s (*Handover*), le nombre de paquets perdus (*Dropped*), le nombre de BU envoyés (*BU*) et la bande passante moyenne en Mbps (*bandwidth*).

**Tableau A.1 Simulations du protocole SMIPv6 avec un usager
qui suit son profil à 100%**

Ld	pckt_rate	speed	intersec	Handover	Dropped	BU	Bandwidth
----	-----------	-------	----------	----------	---------	----	-----------

100	0.1	27.8	5	0.138158	0	23	0.008964
100	0.02	27.8	5	0.059472	0	23	0.04512
100	0.01	27.8	5	0.049675	0	23	0.090305
100	0.007	27.8	5	0.04663	0	23	0.129021
100	0.005	27.8	5	0.044635	0	23	0.180647
100	0.004	27.8	5	0.043583	0	24	0.225855
2	0.004	27.8	5	0.010946	0	23	0.226091
10	0.004	27.8	5	0.014287	0	23	0.226066
30	0.004	27.8	5	0.021007	0	23	0.226032
50	0.004	27.8	5	0.027299	0	23	0.225977
70	0.004	27.8	5	0.033169	0	23	0.22593
90	0.004	27.8	5	0.039733	0	23	0.225863
100	0.004	27.8	5	0.043583	0	24	0.225855
100	0.004	27.8	5	0.043583	0	24	0.226333
100	0.004	27.8	10	0.044082	0	23	0.226336
100	0.004	27.8	15	0.043185	0	24	0.226337
100	0.004	2.8	5	0.043892	0	23	0.226153
100	0.004	13.9	5	0.044049	0	23	0.226021
100	0.004	27.8	5	0.043256	0	23	0.225855
100	0.004	41.7	5	0.044197	0	23	0.22569
100	0.004	55.6	5	0.044211	0	39	0.225544
100	0.004	69.5	5	0.0452005	0	39	0.2253835
100	0.004	83.4	5	0.04619	0	39	0.225223

**Tableau A.2 Simulations du protocole SMIPv6 avec un usager
qui suit son profil à 90%**

Id	pckt_rate	speed	intersec	Handover	Dropped	BU	Bandwidth
100	0.1	27.8	5	0.138689	0	40	0.008964
100	0.02	27.8	5	0.058558	0	40	0.04512
100	0.01	27.8	5	0.048797	0	40	0.090305
100	0.007	27.8	5	0.045615	0	40	0.129021
100	0.005	27.8	5	0.043987	0	40	0.180647
100	0.004	27.8	5	0.043022	0	41	0.225855
2	0.004	27.8	5	0.010295	0	40	0.226091
10	0.004	27.8	5	0.013576	0	40	0.226066
30	0.004	27.8	5	0.02002	0	40	0.226032
50	0.004	27.8	5	0.026732	0	40	0.225977
70	0.004	27.8	5	0.033051	0	40	0.22593
90	0.004	27.8	5	0.039015	0	40	0.225863
100	0.004	27.8	5	0.043022	0	41	0.225855
100	0.004	27.8	5	0.043022	0	41	0.226333
100	0.004	27.8	10	0.043411	0	40	0.226336

100	0.004	27.8	15	0.04235	0	41	0.226337
100	0.004	2.8	5	0.042897	0	205	0.226178
100	0.004	13.9	5	0.043517	0	58	0.226046
100	0.004	27.8	5	0.044456	5	40	0.225855
100	0.004	41.7	5	0.043217	0	34	0.225715
100	0.004	55.6	5	0.066024	51	82	0.225065
100	0.004	69.5	5	0.070655	238	57	0.2246135
100	0.004	83.4	5	0.075286	425	33	0.224162

**Tableau A.3 Simulations du protocole SMIPv6 avec un usager
qui suit son profil à 70%**

Id	pckt_rate	speed	intersec	Handover	Dropped	BU	Bandwidth
100	0.1	27.8	5	0.138183	0	68	0.008964
100	0.02	27.8	5	0.057996	0	68	0.04512
100	0.01	27.8	5	0.048234	0	68	0.090305
100	0.007	27.8	5	0.045223	0	68	0.129021
100	0.005	27.8	5	0.043594	0	68	0.180647
100	0.004	27.8	5	0.042489	0	69	0.225855
2	0.004	27.8	5	0.009875	0	68	0.226091
10	0.004	27.8	5	0.013344	0	68	0.226066
30	0.004	27.8	5	0.019523	0	68	0.226032
50	0.004	27.8	5	0.026266	0	68	0.225977
70	0.004	27.8	5	0.032659	0	68	0.22593
90	0.004	27.8	5	0.039093	0	68	0.225863
100	0.004	27.8	5	0.042489	0	69	0.225855
100	0.004	27.8	5	0.042489	0	66	0.226333
100	0.004	27.8	10	0.042989	0	65	0.226336
100	0.004	27.8	15	0.042231	0	65	0.226324
100	0.004	2.8	5	0.04269	0	472	0.226153
100	0.004	13.9	5	0.042278	0	112	0.226021
100	0.004	27.8	5	0.042471	0	65	0.225855
100	0.004	41.7	5	0.043415	1	50	0.225682
100	0.004	55.6	5	0.086585	100	141	0.224566
100	0.004	69.5	5	0.0809355	280	93	0.224162

100	0.004	83.4	5	0.075286	461	46	0.223758
-----	-------	------	---	----------	-----	----	----------

**Tableau A.4 Simulations du protocole SMIPv6 avec un usager
qui suit son profil à 50%**

Id	pckt_rate	speed	intersec	Handover	Dropped	BU	Bandwidth
100	0.1	27.8	5	0.159164	2	128	0.008952
100	0.02	27.8	5	0.06812	5	128	0.045086
100	0.01	27.8	5	0.064887	16	128	0.090204
100	0.007	27.8	5	0.057403	18	129	0.128919
100	0.005	27.8	5	0.062149	36	128	0.180446
100	0.004	27.8	5	0.057485	37	129	0.225649
2	0.004	27.8	5	0.008358	0	128	0.2261
10	0.004	27.8	5	0.011338	0	128	0.226068
30	0.004	27.8	5	0.018782	0	128	0.226045
50	0.004	27.8	5	0.025016	2	128	0.225968
70	0.004	27.8	5	0.037084	14	129	0.225842
90	0.004	27.8	5	0.049075	26	128	0.22572
100	0.004	27.8	5	0.057485	37	129	0.225649
100	0.004	27.8	5	0.057485	37	126	0.226124
100	0.004	27.8	10	0.041727	0	126	0.226333
100	0.004	27.8	15	0.040613	0	126	0.22632
100	0.004	2.8	5	0.041824	0	1024	0.226135
100	0.004	13.9	5	0.041083	0	227	0.226003
100	0.004	27.8	5	0.058235	38	125	0.225649
100	0.004	41.7	5	0.073105	72	92	0.225138
100	0.004	55.6	5	0.120796	181	133	0.223758
100	0.004	69.5	5	0.160739	337	142	0.2218075
100	0.004	83.4	5	0.200682	493	151	0.219857

Tableau A.5 Simulations du protocole MIPv6

Id	pckt_rate	speed	intersec	Handover	Dropped	BU	Bandwidth
100	0.1	27.8	5	0.391197	31	277	0.007767
100	0.02	27.8	5	0.380276	196	275	0.038928
100	0.01	27.8	5	0.315618	333	273	0.078216
100	0.007	27.8	5	0.428041	466	279	0.111819
100	0.005	27.8	5	0.286258	523	279	0.157199
100	0.004	27.8	5	0.422723	1107	275	0.194318

2	0.004	27.8	5	0.0981	512	273	0.197455
10	0.004	27.8	5	0.173392	474	277	0.197619
30	0.004	27.8	5	0.238077	561	275	0.195695
50	0.004	27.8	5	0.193062	439	275	0.198177
70	0.004	27.8	5	0.309388	499	275	0.197353
90	0.004	27.8	5	0.272378	563	279	0.196994
100	0.004	27.8	5	0.422723	1107	275	0.194318
100	0.004	27.8	5	0.422723	1107	269	0.194212
100	0.004	27.8	10	0.242503	476	273	0.197328
100	0.004	27.8	15	0.16508	308	275	0.198158
100	0.004	2.8	5	0.175432	327	2711	0.199968
100	0.004	13.9	5	0.265993	559	545	0.198459
100	0.004	27.8	5	0.509528	1107	269	0.194212
100	0.004	41.7	5	0.461048	998	179	0.192126
100	0.004	55.6	5	0.355858	765	140	0.191842
100	0.004	69.5	5	0.810211	1582	108	0.18048
100	0.004	83.4	5	0.568434	1234	92	0.180667

BIBLIOGRAPHIE

- [1] D. Johnson, C.Perkins et J. Arkko, « Mobility Support in Ipv6 », IETF Draft draft-ietf-mobileip-ipv6-21.txt, Février 2003.
- [2] T. Soueid, « Présentation de Mobile Ipv6 », http://membres.lycos.fr/tonysoueid/divers/Mobile_IPv6.pdf, 2002.
- [3] N. Montavont et T. Noël, « Handover Management for Mobile Nodes in IPv6 networks », Communications Magazine, IEEE, Volume : 40, No. : 8, Août 2002, Pages : 38 – 43.
- [4] S.J. Vaughan-Nichols, « Mobile IPv6 and the Future of Wireless Internet Access », Computer, Volume : 36, No. : 2, Février 2003, Pages : 18 – 20.
- [5] H. Soliman, C. Castelluccia, K. El-Malki et L. Bellier, « Hierarchical Mobile IPv6 mobility management (HMIPv6) », IETF Mobile IP Working Group, draft-ietf-mobileip-hmipv6-07.txt, Octobre 2002.
- [6] R. Koodli, « Fast Handovers for Mobile IPv6 », IETF Mobile IP Working Group, draft-ietf-mipshop-fast-mipv6-01.txt, Janvier 2004.
- [7] S. Auvray, « Fast Handovers for Mobile IPv6 », EURESCOM Participants in Project P1113, 2002.
- [8] A. Valko, J. Gomez et A. Campbell, « Cellular IP », IETF, draft-valko-cellularip-00.txt, Novembre 1998.

- [9] K. El Malki et H. Soliman, « Simultaneous Bindings for Mobile IPv6 Fast Handovers », IETF, draft-elmalki-mobileip-bicasting-v6-03-txt, Mai 2003.
- [10] J. Murray, « Hierarchical MIPv6 Mobility Management using simultaneous Bindings for MIPv6 Fast Handoffs », Ericsson Canada Inc, Mai 2002.
- [11] E. Cayirci et I.F. Akyildiz, « User Mobility Pattern Scheme for Location Update and Paging in Wireless Systems », Mobile Computing, IEEE Transactions on, Volume : 1, No. : 3, Juillet - Septembre 2002, pages : 236 – 247.
- [12] N. Frangiadakis, M. Kyriakakos, S. Hadjiefthymiades et L. Merakos, « Realistic Mobility Pattern Generator : Design and application in path prediction algorithm evaluation », Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on, Volume : 2, 15-18 Septembre 2002, pages : 765 – 769.
- [13] G. P. Pollini et Chih-Lin I « A profile-Based Location Strategy and its performance », Selected Areas in Communications, IEEE Journal on, Volume : 15, No. : 8, Octobre 1997, pages : 1415 – 1424.
- [14] T. Yang, Y. Dong, B. Zhou, D. Makrakis, « Profile-Based Mobile MPLS Protocol », Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on, Volume : 3, 12-15 Mai 2002, pages : 1352 – 1356.
- [15] G.Y. Liu; G.Q. Jr Maguire, « A predictive mobility management algorithm for wireless mobile computing and communications », Universal Personal

Communications. 1995. Record., 1995 Fourth IEEE International Conference on, 6-10 Novembre 1995, pages : 268 – 272.

- [16] P. McCann, «Mobile IPv6 Fast Handovers for 802.11 Networks», IETF Mobile IP Group, Février 2004.
- [17] T. Ernst, «MobiWan: A NS-2.1b6 simulation platform for Mobile IPv6 in Wide Area Networks», Motorola Labs Paris, Juin 2001.
- [18] I. Vivaldi, B.M Ali, H. Habaebi, V. Prakash et A. Sali, « Routing Scheme for Macro Mobility Handover in Hierarchical Mobile IPv6 Network », Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on, 14-15 Jan. 2003, pages : 88 – 92.
- [19] C.H. Chu et C.M. Weng, « Pointer Forwarding MIPv6 Mobility Management », Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, Volume : 3, 17-21 Novembre 2002, pages : 2133 – 2137.
- [20] K. Omae, T. Ikeda, M. Inoue, I. Okajima et N. Umeda, « Mobile Node Extension Employing Buffering Function to Improve Handoff Performance », Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on, Volume : 1, 27-30 Octobre 2002, pages : 62 – 66.
- [21] L. Patanapongpibul et G. Mapp, « A client-based handoff mechanism for mobile IPv6 wireless networks », Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on, 30 Juin-3 Juillet 2003, pages : 563 - 568.

- [22] S. Thomson et T. Narten, « IPv6 Stateless Address Autoconfiguration », RFC 2462, Décembre 1998.
- [23] G. Chakraborty, B. Bista, D. Chakrabort et N. Shiratori, « Location Management in PCN by Movement Prediction of the Mobile Host », Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on, Volume : 1, 8-11 Juillet 2002, pages : 78 – 83.
- [24] R. Hinden et S. Deering, « IP Version 6 Addressing Architecture », RFC 2373, Juillet 1998.
- [25] J. Arkko, G. Kuijpers, H. Soliman, J. Loughney et J. Wiljakka, « Internet Protocol Version 6 (IPv6) for Some Second and Third Generation Cellular Hosts », RFC 3316, Avril 2003.